NASA CONTRACTOR REPORT

NASA CR-2238

N73-22055

CASE FILE COPY

NASA CR-2238

STUDIES TO DESIGN AND DEVELOP IMPROVED REMOTE MANIPULATOR SYSTEMS

by J. W. Hill and A. J. Sword

Prepared by
STANFORD RESEARCH INSTITUTE
Menlo Park, Calif.
for Ames Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1973

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-2238		

4. Title and Subtitle	5. Report Date
"Studies to Design and Develop Improved Remote Manipulator Systems"	April 1973
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
J. W. Hill and A. J. Sword	
	10. Work Unit No.
9. Performing Organization Name and Address	
Stanford Research Institute Menlo Park, California	11. Contract or Grant No. NAS 2-6680
	13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address	
National Aeronautics & Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This report describes a philosophy of remote manipulator control based upon several levels of automatic supervision which derives manipulator commands from an analysis of sensor states and the task requirements.

Principle sensors are manipulator joint position, tactile, and motor currents. The tactile sensor states can be displayed visually in perspective or replicated in the operator's control handle or perceived by the automatic supervisor.

Studies are reported on control organization, operator performance and system performance measures. Unusual hardware and software details are described.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Teleoperator Robot Automatic Control Man-Machine	UNCLASSIFIED-UNLIMITED

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
UNCLASSIFIED	UNCLASSIFIED	156	3.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

Page Intentionally Left Blank

FOREWORD

This report was prepared by Stanford Research Institute under Contract NAS2-6680, monitored by James L. Jones, Ames Research Center, National Aeronautics and Space Administration, Moffett Field, California. John W. Hill was project leader.

We would like to acknowledge the contributions of Professors William R. Verplank and James L. Adams of Stanford University, who suggested changes to the PICKUP program to make it more reflexive, and the design of the supervisory control experiment, respectively; Carroll M. Steel, who designed the tactile-computer interface; Norman T. Hopper, who designed the arm modifications, hand controller, and range finder; J. R. Woodbury, who designed and built the range finder electronics; and Anthony F. Ferrera, who helped develop and construct electronic interfaces for several systems peripheral to the computer.

ABSTRACT

This report describes a philosophy of automatic control based on both coded and analog information and shows how to construct and communicate with a simple arm automation by using four increasingly automatic levels of supervisory control. The communication uses a natural language consisting of both coded and analog information to carry out tasks with a seven-degrees-of-freedom manipulator. The supervisory control is largely based on information from touch sensors mounted on the end effector. The use of tactile information in manipulation is studied in two ways. First, such information conveyed directly to the hand of the human operator allows him to be more efficient, avoiding drops and fumbles, and allows him to perform--with poor or restricted vision--tasks that could not otherwise be carried out. Second, tactile information provided to the computer controller enables the mechanical arm to react with simple automatic reflexes and to act with automatic grasping abilities.

Manipulation with a transmission delay simulation is studied using an on-line computer to measure movement and waiting times. The simple move-and-wait strategy previously reported by Sheridan and Ferrell (1963)* is found not to be followed; a more complex sequence of events actually occurs. A preliminary compensatory tracking evaluation showed that a teleoperator can be characterized in terms of simple changes in the operator-teleoperator describing function and that corresponding figures of merit can be obtained.

---

*References are listed at the end of the report.

CONTENTS

ILLUSTRATIONS

TABLES

I INTRODUCTION

When we leave the laboratory environment where master-slave manipulators are king to perform manipulation tasks in the field, we must either pay dearly for a bilateral (force-reflecting) manipulator or accept an order of magnitude increase in task completion time from a unilateral system. In distant space operations involving transmission delays, force reflection actually impedes performance, and we are forced to use the slower unilateral manipulator. If we equip the remote arm with a minicomputer and some sensors, however, man's reflexive actions and adeptness can be recreated at the remote scene, drastically reducing task completion time. The decision-making ability of a small computer, coupled with its ability to direct stored and practiced sequences of motions, enable remote tasks to be carried out quickly without the necessity of feeding intermediate force information back to to the operator or requiring intermediate control information from him.

Other advantages of computer-augmented teleoperator* control are evident at the control station, where a small control computer can generate displays, calculate difficult coordinate transformations, and allow man to alternately exercise control with joysticks, push-buttons or typed instructions. Eventually the computer will help him plan and carry out the best strategy for completing a given job in the shortest time.

---

*Teleoperator is defined as a general purpose, dexterous, cybernetic machine that communicates man's bodily dexterity across a barrier to mechanical actuators (1) that can operate under loads too great for any unaided man, (2) in an environment too hostile or too far away, or (3) in the case of prosthetics, that help a handicapped man become more nearly normal (Johnson and Corliss, 1967).

A primary goal of our research effort over the past few years has been the design of a computer-augmented teleoperator control scheme that optimizes performance in carrying out remote tasks by combining the best attributes of man and state-of-the-art computing, Man's ability to interpret scenes, estimate distances, and project motion with a multicoordinate control brace is combined with the computer's ability to save and accurately duplicate arm positions, remember sequences of motions, carry out tests based on arm positions, and interpret touch sensors. General background material on such supervisory control systems has been summarized by Johnson and Corliss (1967) and Corliss and Johnson (1969).

In the course of this research, we have developed an experimental facility consisting of computer-augmented local and remote stations coupled through a transmission delay. The operating system consists of a unique interactive control language that permits manual and automatic control modes to be used separately or in combination. Of prime importance in this work has been the development of a touch-sensing system for the remote end effector. These sensors are the source of information for both sensory displays and automatic control modes. The results of this research appear in two final reports (Bliss, Hill, and Wilber, 1970; and Hill and Bliss, 1971a); in two papers outlining the supervisory control concept (Hill and Bliss, 1971b; and Hill and Sword, 1972); and in a paper focusing on automatic control of a hand prosthesis (Hill, 1972) reproduced in Appendix I.

This report covers a one-year research effort on additional work in this area. Objectives of this additional work include the further development of a computer-aided teleoperator control system, tasks and measurement techniques for evaluating human manipulation performance, and remote sensing and display techniques. The body of this report covers the work toward these objectives. The nine appendices describe techniques

and instrumentation necessary for computer-augmented teleoperator control and performance measurement.

Section II describes the computer-simulated supervisory control system. New additions to the previous control system (Hill and Bliss, 1971) are (1) the Computek control console, which, because of its high speed, allows the computer to communicate quickly with words instead of abbreviated symbols and letters used previously, and (2) a tactile interface to the computer, which allows the state of the touch sensors to be read quickly and used to control actions and generate displays.

Section III describes the control modes of the supervisory control system. New developments include an interactive decision-response control mode with roughly double the numbers of commands and tests. In this mode the computer communicates with the operator in sentence-like statements identical to those of the ARM (algorithmic remote manipulation) language. The ARM language itself has been expanded to include both special and general sensor tests. The general tests permit any of the millions of possible combinations of activated external sensors or activated internal jaw sensors to be specified by the operator and tested in a single command. In addition, the concept of a new control mode (DYNARM) for concatenating ARM programs to specify a complete task is described.

Section IV reports the development of an on-line technique for measuring human performance in a time-delay situation. The computer is programmed to measure accurately the number and duration of the operator's moves and pauses during a task. Using this performance monitor in a time-delayed manipulation task, we found that while humans do move, wait, move, wait, ..., etc., they do not use Sheridan and Ferrell's (1963) move-and-wait strategy, but rather move in a more complex manner. Move-while-moving and multiple-move strategies are frequently used.

Section V describes the completed design of two displays that provide remote touch information to the operator. Completing the brace-mounted tactile display on the operator's hand involved the design of stimulators corresponding to the external hand sensors (excluding fingers) and a smoothly acting jaw controller that did not interface with the bimorph finger displays. A computer-generated scope display of the touch information includes motion of the jaw and, in addition, gives information of the object's size.

The compensatory tracking analysis of Section VI provides initial describing function data for the Rancho teleoperator. A combination of tracking runs using the human arm, Rancho brace, and Rancho arm as the controlled vehicle, permitted limitations of both brace and mechanical arm to be determined separately. Remnant data indicate that this teleoperator accentuates low frequency operator noise and filters out high frequency operator noise. The concept of a teleoperator-induced remnant was unexpected; that of a teleoperator-induced filter was expected. Figures of merit for the Rancho system in terms of open loop gain reduction, increased time delay (also known as the "critical" time), and internal noise were easy to characterize from the data.

Finally, the appendices describe a new range sensor, computer techniques, and experimental designs.

II ARM CONTROL SYSTEM

The control system for carrying out tasks at a distant location with a mechanical arm consists of the following three basic elements:

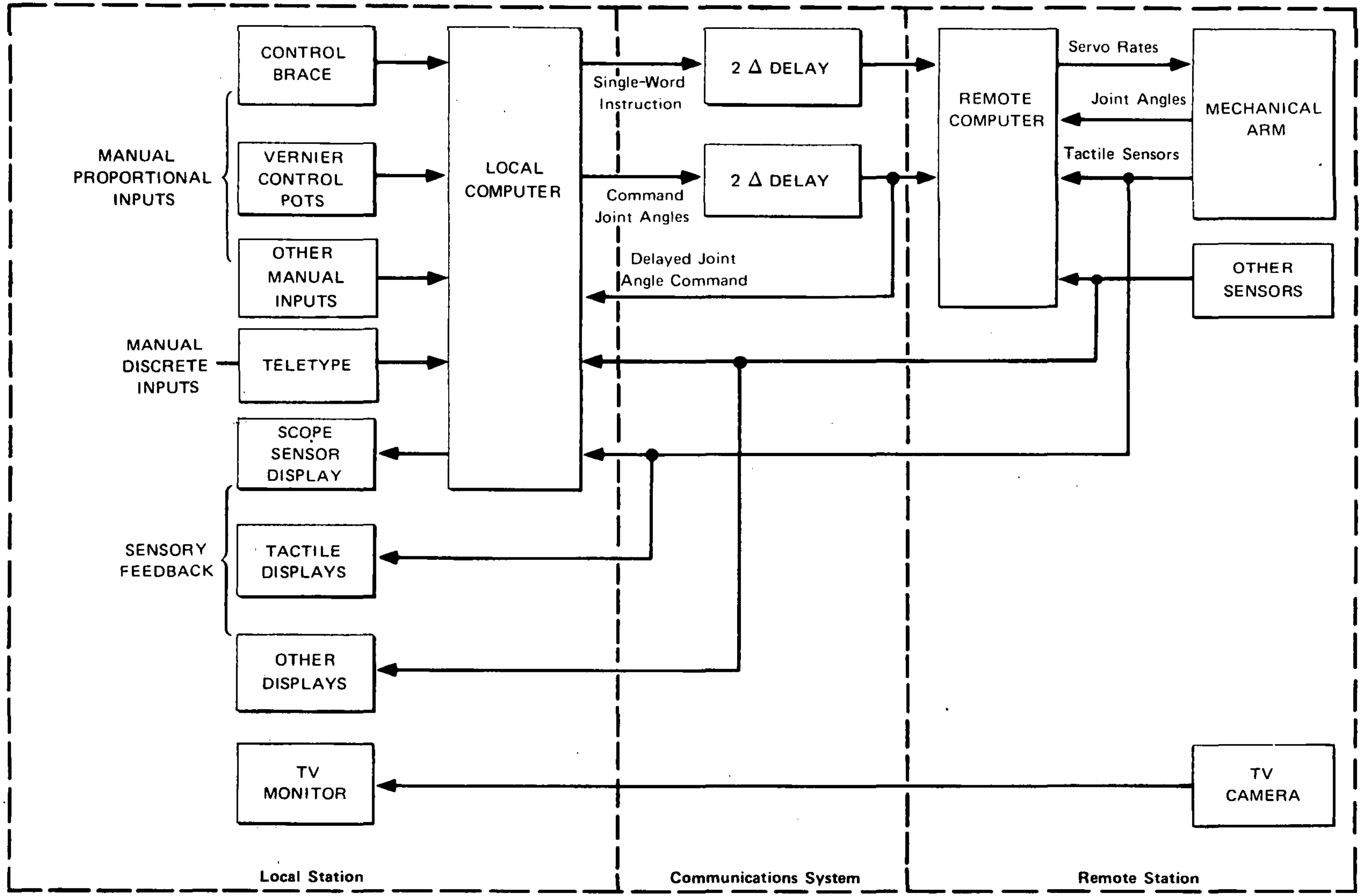
- A control station where the operator controls the motion of the arm by transmitting commands in the form of both analog and/or discrete information and by supervising the resulting action using various displays and feedback.
- A remote station that accepts the commands and uses them along with information from environment sensors to control the arm.
- A communications link that limits information flow. The limitations may take the form of a time delay, a bandwidth limitation, a signal-to-noise ratio, maximum video frame rate, etc.

A. Control Station

The arm control station is shown schematically in the left half of Figure 1. It consists of several manual inputs, several visual and tactile displays, and a teletype input to a computer in order to permit the operator to select and transfer among inputs, displays, and programs of motion to accomplish a given task.

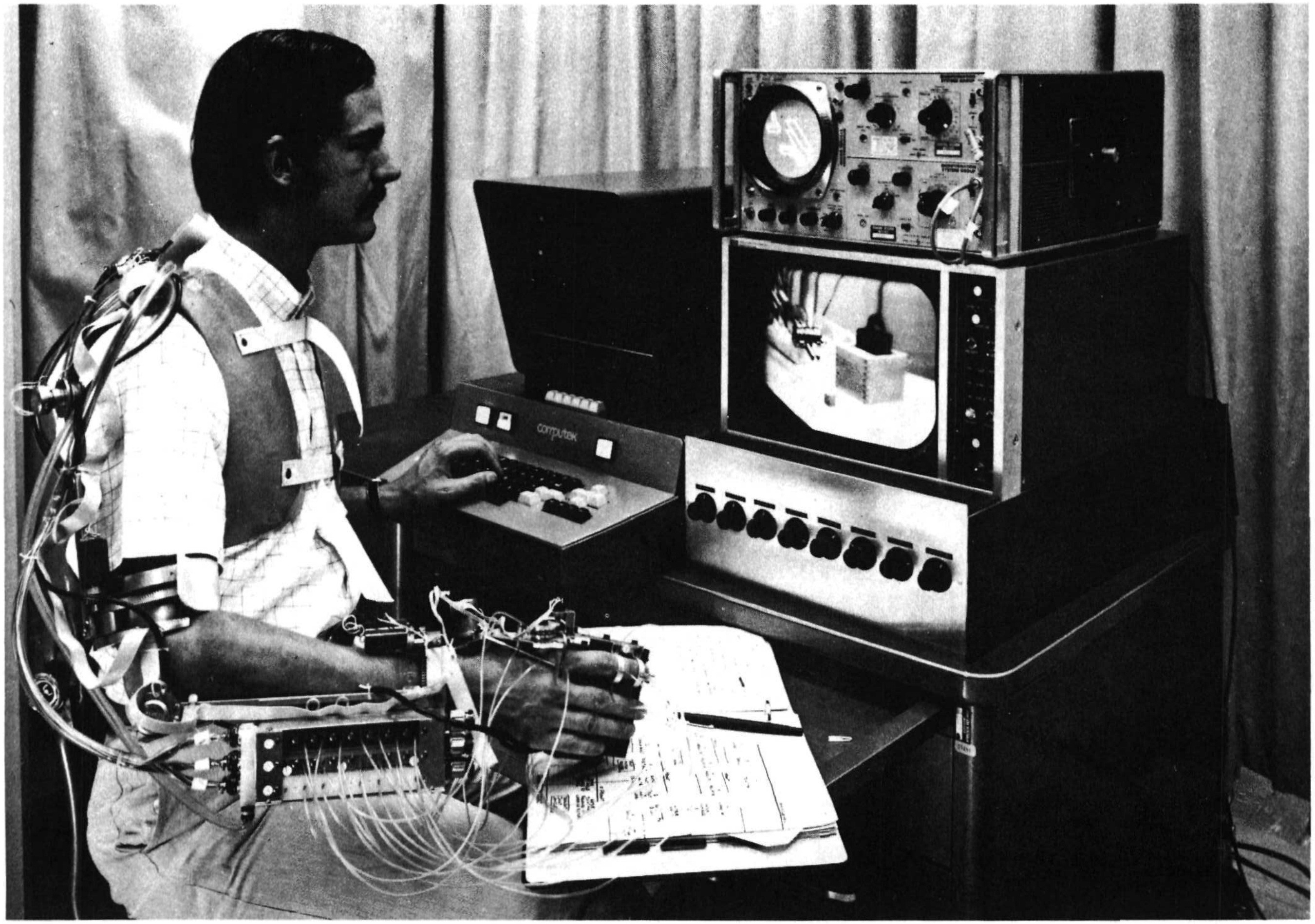
1. Manual Inputs

The manual inputs are illustrated in Figure 2. The Rancho anthropomorphic control brace measures the joint angles of the operator's arm with a set of seven potentiometers. These joint angles can also be controlled with individual potentiometers mounted on a panel. Manipulations in tasks requiring either a long time to complete or precise positioning are generally best carried out with knob control.



TA-720531-19R

FIGURE 1 BLOCK DIAGRAM OF THE MAN-MACHINE CONTROL SYSTEM



TA-760522-8

FIGURE 2 ARM CONTROL STATION. Top left, Teletype terminal; bottom left, master brace with 6 x 24 tactile displays on thumb and index finger; top right, computer-driven display with sensing pad information; middle right, TV view of hand and work space; bottom right, bank of eight vernier potentiometers. Operator is issuing a teletype command.

Control can also be entered directly from a teletype, using the format #6 = 45 where "#" and "=" are prompts from the teletype, "6" is the joint number, and "45" is the joint angle in degrees. Teletype control has been used largely for testing and the debugging of manipulation programs. It would be desirable to add a joystick-type controller and perhaps a miniature (scaled down) control brace light enough to maintain the position it is put in and small enough to be manipulated with the operator's fingers. These additional manual inputs would greatly enhance the flexibility of the local station and enable the operator to exercise greater freedom in planning strategy to complete a task. In addition, comparative performance studies could be run to determine what type of control is most effective for a specific type of task.

2. Sensory Feedback

Primary feedback is supplied by a closed-circuit television system. In addition, a computer-driven scope display presents the state of the touch-sensor information. This display is described in Section V. Here information from tactile sensors, as well as the degree of jaw closure, are presented in perspective.

Tactile feedback from the arm to the operator is provided by a set of touch sensors on the hand. The touch sensors fall into two groups: (1) a pair of touch-sensing pads on the gripping surface of the manipulator tongs and (2) a number of individual contact sensors covering the outer surface of the tongs and wrist. Since these sensors and the anthropomorphic tactile stimulators represent a major part of this project, they are separately described in Section V.

B. Remote Station

The remote station as it currently stands in the communication system is shown schematically in the right half of Figure 1. It consists of a modified Rancho arm* with a number of specially built touch sensors, a TV camera, and a computer. The physical layout is shown in Figure 3.

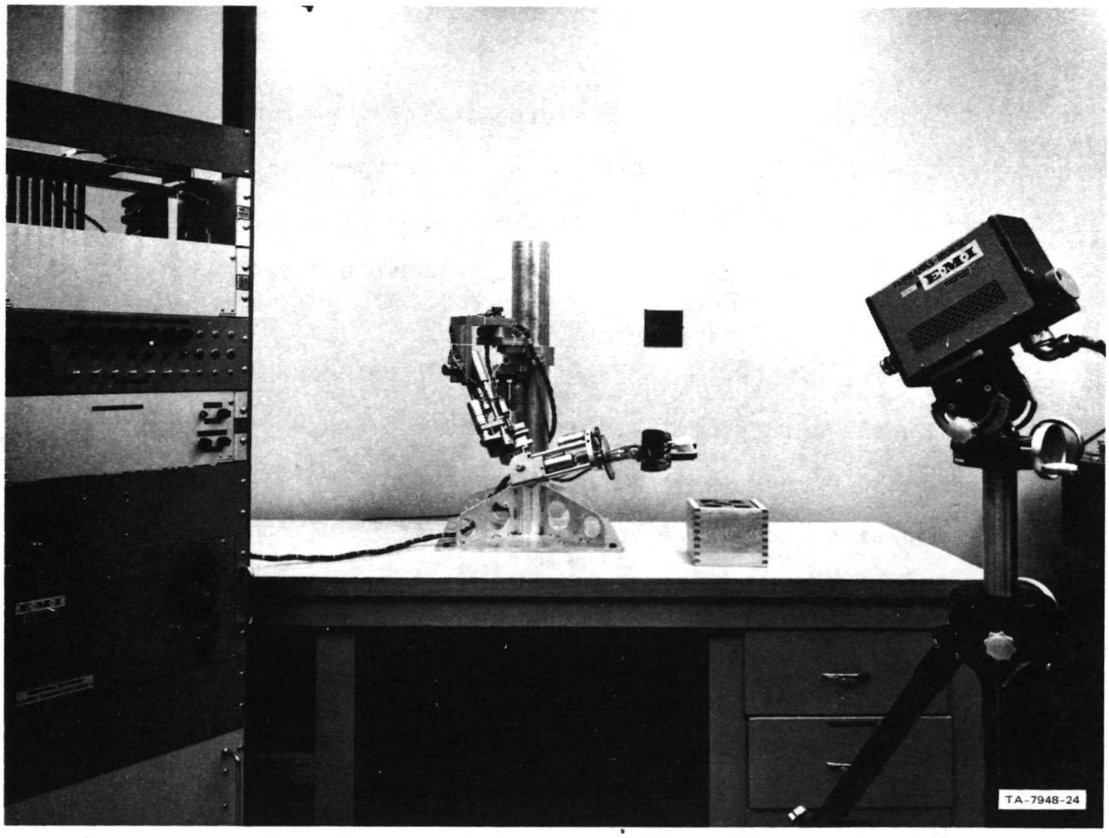


FIGURE 3 REMOTE STATION. From left to right are (1) the computer interface containing proportional control amplifiers, sensor amplifiers, and power supplies, (2) the modified Rancho Arm, and (3) the TV camera.

---

*Model 8A, a seven-degrees-of-freedom anthropomorphic manipulator manufactured by R&D Electronics, Downey, California.

1. Mechanical Arm

During this study, the Rancho arm was upgraded to reduce the amount of play in the joints and to increase the range of motion. In total, all joints but one have been refurbished to some degree, two members have been completely replaced, and two joints completely rebuilt. These changes were deemed necessary, based on our previous experience with the arm, in order to carry out meaningful manipulation experiments with it.

An initial study of the sources of play or looseness in the Rancho arm revealed that poor design in the three worm-gear-driven joints was the major trouble. Replacing the bearings with commercial roller bearings and incorporating simple backlash adjusters greatly improved smoothness of performance. To lighten the arm, cable drivers for jaw closure and wrist prehension/extension were lengthened in order to mount the motors on the main pedestal. To extend the range of hand motion, thus raising the number of meaningful tasks that could be carried out, wrist flexion/extension range was increased from 90° to 180°, and wrist rotation (supination/pronation) was increased from 90° to 360°. To reduce the play between the tongs, the prehension linkages were rebuilt; the machine-screw bearings were replaced with tightly fitting pin bearings.

2. Proportional Arm Controller

Because of the many difficulties experienced with the original relay-operated bang-bang control system, a new proportional control system was designed and built. The new system has the following advantages over the original system:

- The time for a given movement can be halved by driving the motors harder than the original system while still retaining stability.

- Smaller movements are permissible.
- The smooth acceleration and deceleration reduces the mechanical coupling between joints and the vibrations at the beginning and termination of movements.
- Proportional control allows computer programs to govern rates of motion as well as position.

The proportional power amplifiers use a pulse-width-modulation drive to keep the power dissipated in the drive transistors low and also to limit the drive current to prevent the motors from burning out. Since torque is proportional to motor current, this current limiting also provides a linear and easily adjustable analog to a mechanical clutch.

3. Sensors

There are two types of tactile sensors presently on the mechanical arm. Both are constructed in similar fashion. The external sensors indicate when the outside of the tongs or hand come into contact with any object. The grab sensors indicate the shape of whatever object the hand is grasping. These sensors are more fully discussed in Section V.

In addition to the above sensors, work is presently being done under separate NASA contract* to develop a complete hand sensing system composed of proportional grab sensors, area contact sensors, and slip detector sensors. These sensors will overcome the present limitations of sensor durability. It is hoped that when the design is complete, a set can be installed on the Rancho arm to replace those that presently exist.

---

*This research is being carried out under Contract SNSN-63 from the National Aeronautics and Space Administration.

Primary feedback is provided by a television camera, located at the remote station, that provides the operator with a view of the entire arm and its work space. Recently, a second television camera has become available, and plans are to mount this camera on the pedestal that supports the arm, thus providing the operator with a closeup view of the work space from above. This corresponds to the view the operator would have if he were performing the task in person. This second camera also allows an evaluation to be performed to determine effective placement of the cameras in order to enhance operator performance without confusing him by presenting him with two different television pictures.

Recently, a design has been completed and some of the parts fabricated for a range sensor. This sensor will give the operator information that was previously unavailable. He will be able to determine if he is about to touch an object, and if so, he can determine the approximate distance to and direction of the object. Further, the operator will be able to determine if the hand is properly positioned prior to grasping the object. This will be extremely useful in situations where, due to the position of the television cameras and the alignment of the hand, the operator's view is obscured. The range sensor is shown mounted on the hand in Figure 4. The construction and operation of this sensor is discussed in detail in Appendix A.

C. Limited Communications Link

The limited communications link has been incorporated into the system in order to provide the ability to investigate realistic situations in which hazardous conditions exist, large distances are involved, or both. The limitations may take the form of:

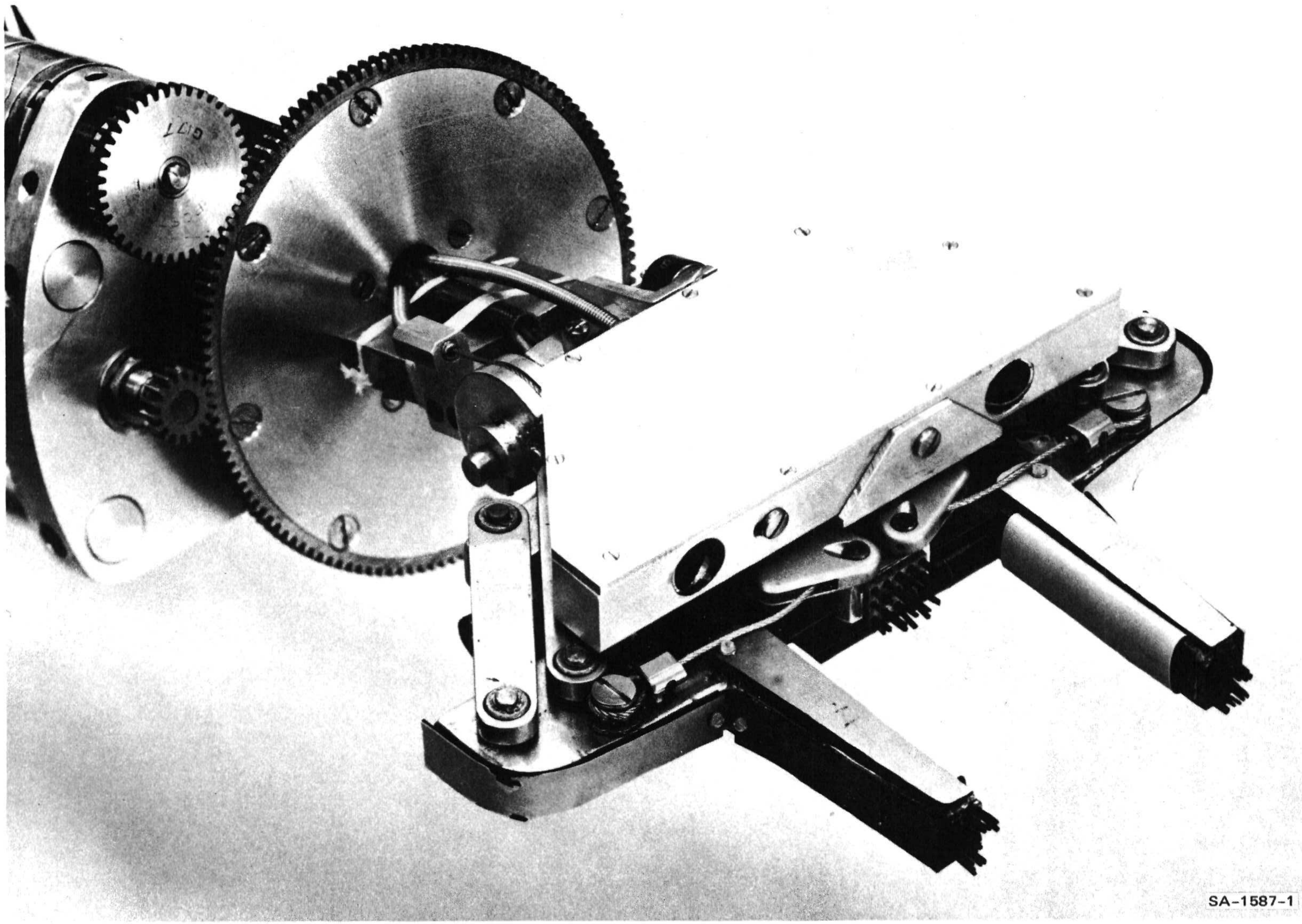


FIGURE 4 RANGE SENSOR MOUNTED ON HAND

- Bandlimiting
- Noise
- Time delay.

The first two of these limitations involve the television picture and can be introduced into the system by adjusting the television camera/monitor system. The transmission delay limitation has been simulated by computer. The delay available is variable from zero to twenty minutes in 1/30th second increments. The delay operates by constructing a queue, called a delay line, in core and disk memory.

All analog and discrete commands are inserted into one cell of the delay line at the same time that delayed commands to be executed are obtained from the cell, as shown in Figure 5. First, the delayed command is removed from the ith cell, and then the most recent command is inserted into the same cell. For the next cycle, the i + 1st cell is used. Since this process occurs at the rate of 30 per second, each cell represents 1/30th of a second. Thus, the total delay is equal to the number (N) of cells in the delay line times 30.

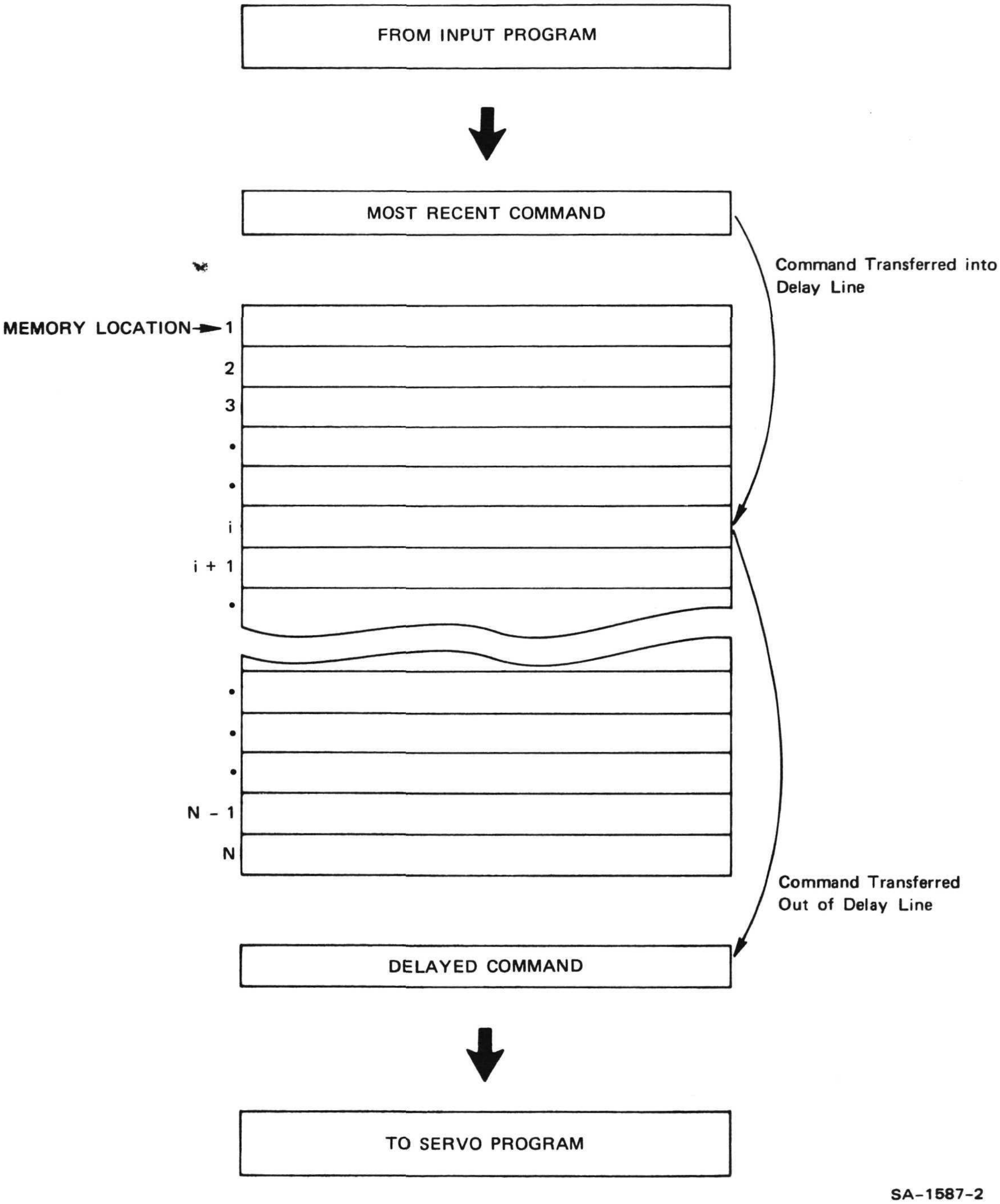


FIGURE 5 EFFICIENT SIMULATION OF TRANSMISSION DELAY
First the delayed command is removed from the i^{th} cell, and then the most recent command is inserted into the same cell. For the next cycle, the $i + 1^{st}$ cell is used.

III ARM CONTROL MODES

In order to interact with the control system described in the previous section, three different control levels have been implemented: (1) manual, (2) decision-response, and (3) ARM (algorithmic remote manipulation) language. Two other control modes are in various stages of planning: (1) DYNARM (dynamic arm programmer) and (2) the planner. These control levels span the continuum between purely manual control and complete machine control. Varying amounts of computer-augmented assistance can be called on to match the certainty or uncertainty of the task. The primary advantage of the multilevel manipulation is that the operator is at liberty to select any desired control mode and then to intervene at will, perhaps in mid-task, and change to any other control mode. With a system of such great flexibility, studies can and have been undertaken to determine what is the most efficient combination of man and machine to complete any specific class of tasks.

An integral part of the above five levels of control is the computer controller located in the remote station. A block diagram of the computer controller is given in Figure 6. An instruction (two 12-bit words) and the analog joint commands (seven 12-bit words) from the control station are the only inputs. Arm control is quite conventional; the actual joint positions (obtained by analog-to-digital conversion) are subtracted from the command joint positions, and this difference is multiplied by the joint gains and then output to the servo motors (via digital-to-analog conversion) to establish angular rates. The transfer register is used to offset the analog commands, so that control can be transferred to the human operator smoothly after an automatic operation has been completed. Thus, with the use of these registers and the memory, in combination with

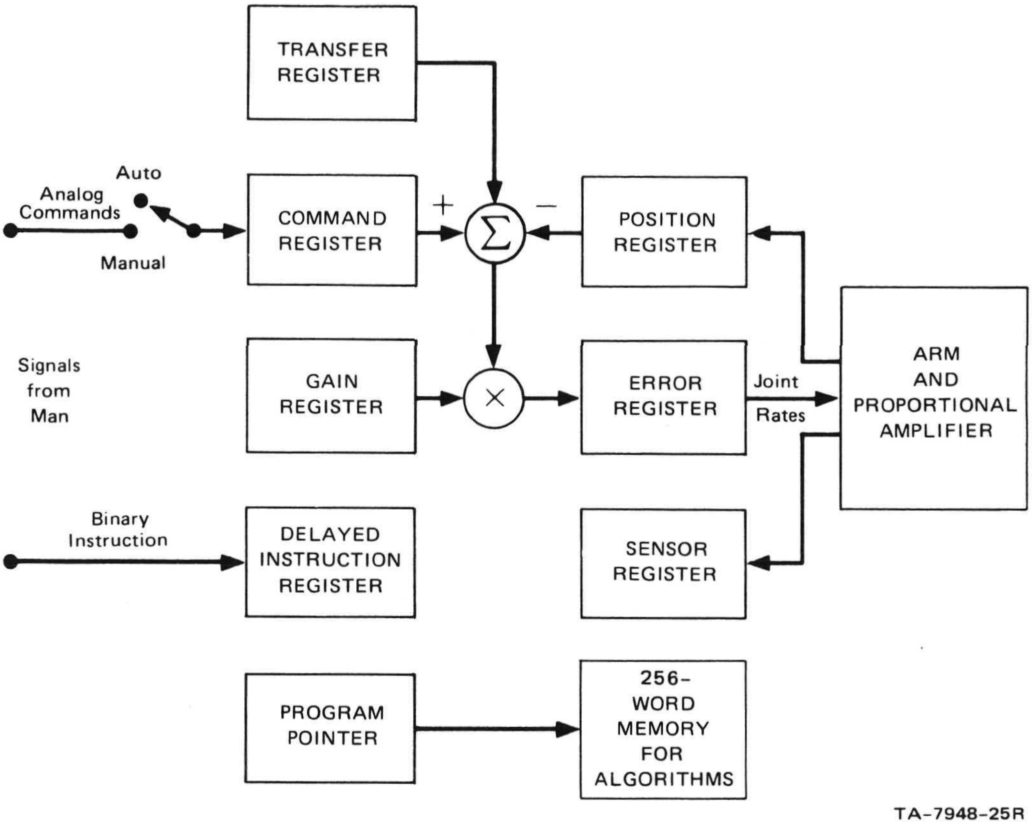


FIGURE 6 COMPONENTS OF THE COMPUTER CONTROLLER

the registers in the control system and the sensor information, meaningful tasks can be carried out.

A. Manual Control

Three different types of manual control modes have presently been implemented into the arm control system: knob, brace, and teletype control. These manual modes have been designed with man's efficiency of operation as the paramount consideration. Knob control is achieved through a bank of seven potentiometers located at the local station. When the operator specifies knob control by typing K (the computer prints

KNOBS*), and further specifies whether control is to be by the absolute values of the potentiometers or by their relative changes by typing A or R (ABSOLUTE or RELATIVE), the computer reads the knob angles as voltages by analog-to-digital (A/D) conversion, and the arm moves to those angles. Brace control (BRACE) is identical to knob control, except that voltages from the potentiometers on a seven-degrees-of-freedom Rancho anthropomorphic brace worn by the human operator are used as command signals. A different set of A/D channels are used for this purpose.

The teletype control mode is used not only as a control mode, but also as an information mode. By typing a T (the computer prints TTY) the operator can enter the teletype mode; he can then specify COMMANDS or POSITIONS, causing either the most recent command vector or the arm position vector to be printed out. The operator may also specify an angle to which a specific joint is to move. These commands and their relationship are illustrated in Figure 7. An example will illustrate how these manual modes are used in conjunction with one another to accomplish a task.

Suppose the operator is faced with the task of picking up a round peg and placing it in a round hole. One of the possible command sequences for such a task is as follows:

```
>BRACE:ABSOLUTE

>KNOBS:RELATIVE

>BRACE:RELATIVE

>KNOBS:RELATIVE
```

---

*Throughout this report, on-line communication with the control computer is represented by capital letters. The underlined letters are those typed on the control typewriter by the operator.

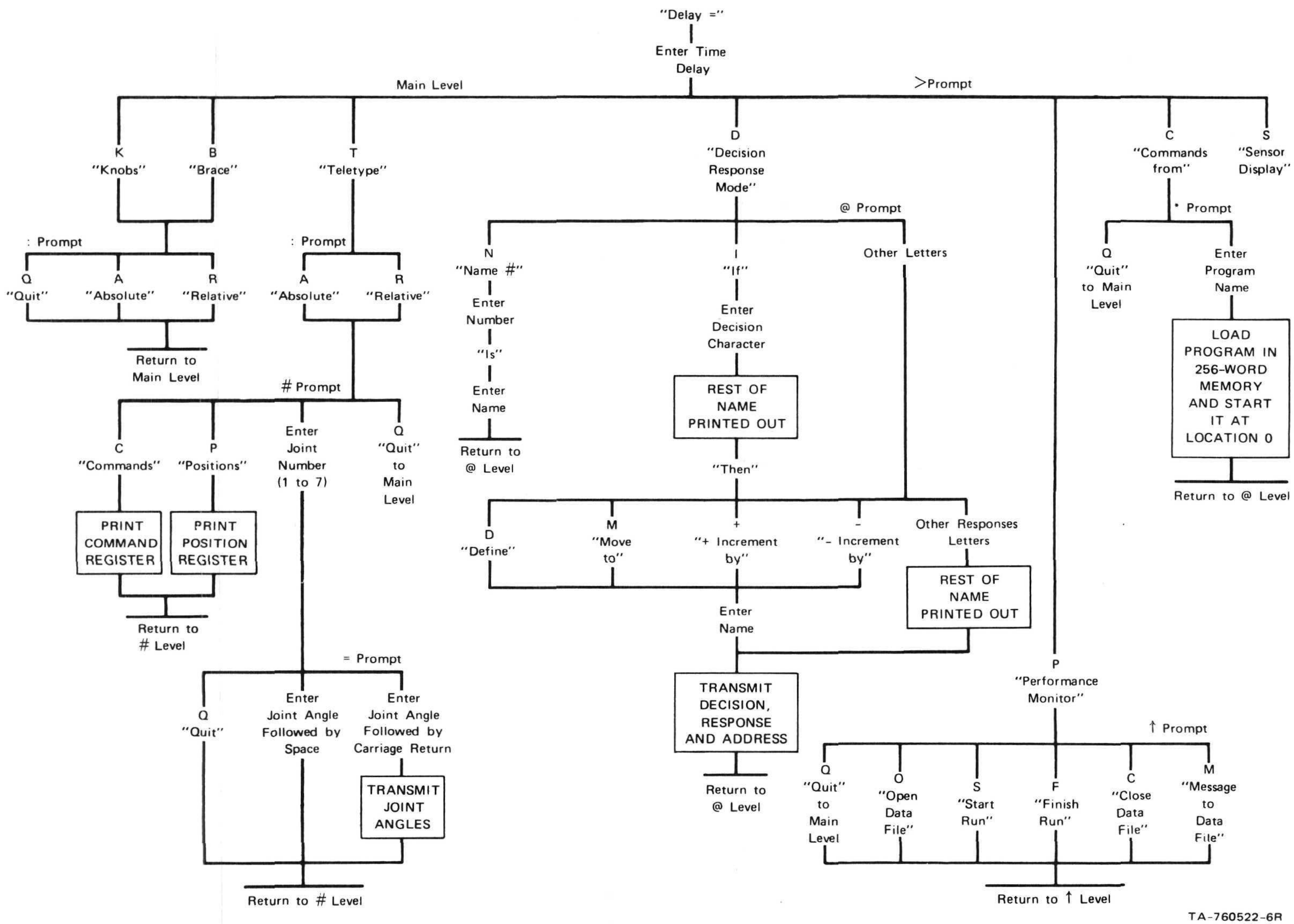


FIGURE 7 COMMAND STRUCTURE FOR THE SUPERVISORY CONTROL STATION

Here the operator uses the BRACE ABSOLUTE command to allow him to move the arm in the vicinity of the peg. He then uses the KNOBS RELATIVE command to transfer control to the knobs without any transient motions. The knobs are used for precise control to pick up the peg. The operator then goes to relative control from the brace and moves to the bin. Finally, he transfers control to the knobs in the relative mode and inserts the peg in the hole. Notice that the operator generally uses the brace for gross arm motions and the knobs for very precise arm motions. Notice also that the teletype mode is not used; it is used primarily for diagnostic work and program writing.

B. Decision-Response Control

Under the decision-response mode of control, the operator has considerably more flexibility than with the manual mode. The decision-response mode is actually a dual one. The operator may specify via teletype a test and an action to be completed, provided that the test is passed. Alternatively, the operator may specify that only the action is to be carried out. Thus, the two aspects of this mode of control may be thought of as being reflexive and commanded.

Examples of this control mode, are:

IF ANYSENSOR THEN CLOSE

or

CLOSE

In the first case, the jaws will close only if a tactile sensor has been activated, whereas in the second case, the jaws will close regardless of

sensor state. Control by this mode may be achieved by responding to the ">" prompting symbol with a D (for DECISION-RESPONSE MODE). The computer then signals that it is ready to accept commands by prompting with "@." This is illustrated in Figure 7. One of the unique features of this system is that the operator may specify the test and action and then manually move the arm. Thus, the operator might specify:

IF FINGERTIP THEN STOP

The operator may then move the arm about, and if the fingertip sensors contact anything, the arm will automatically stop. For the computer, this instruction is coded by FS, where the fingertip sensor closed test is specified by F, and the STOP command is specified by S. Appendix H lists all of the currently available tests and actions, together with a description of how they affect the registers of the computer controller.

Discrete instructions transmitted by the human operator are saved in the instruction register. The basic form of the instruction specifies a test and an action, called a decision-response pair, and a numerical parameter. Not shown in Figure 6, but essential to the operation of the automatic controller, is the instruction processor shown in Figure 8. The processor transfers numbers between registers and carries out sensor and position tests on the basis of individual instructions. These instructions are the building blocks used by single commands from the control typewriter. A single instruction requests that a specific test be executed and that a specific command be carried out if the test is passed. The first half of the instruction word (6 bits) is used to select one of 64 possible tests by means of a look-up table. If the test is passed, the second half of the instruction (6 bits) is similarly used to select one of 64 possible actions. Even though only 19 tests and 25 actions have been implemented, a rich variety of operations is already possible.

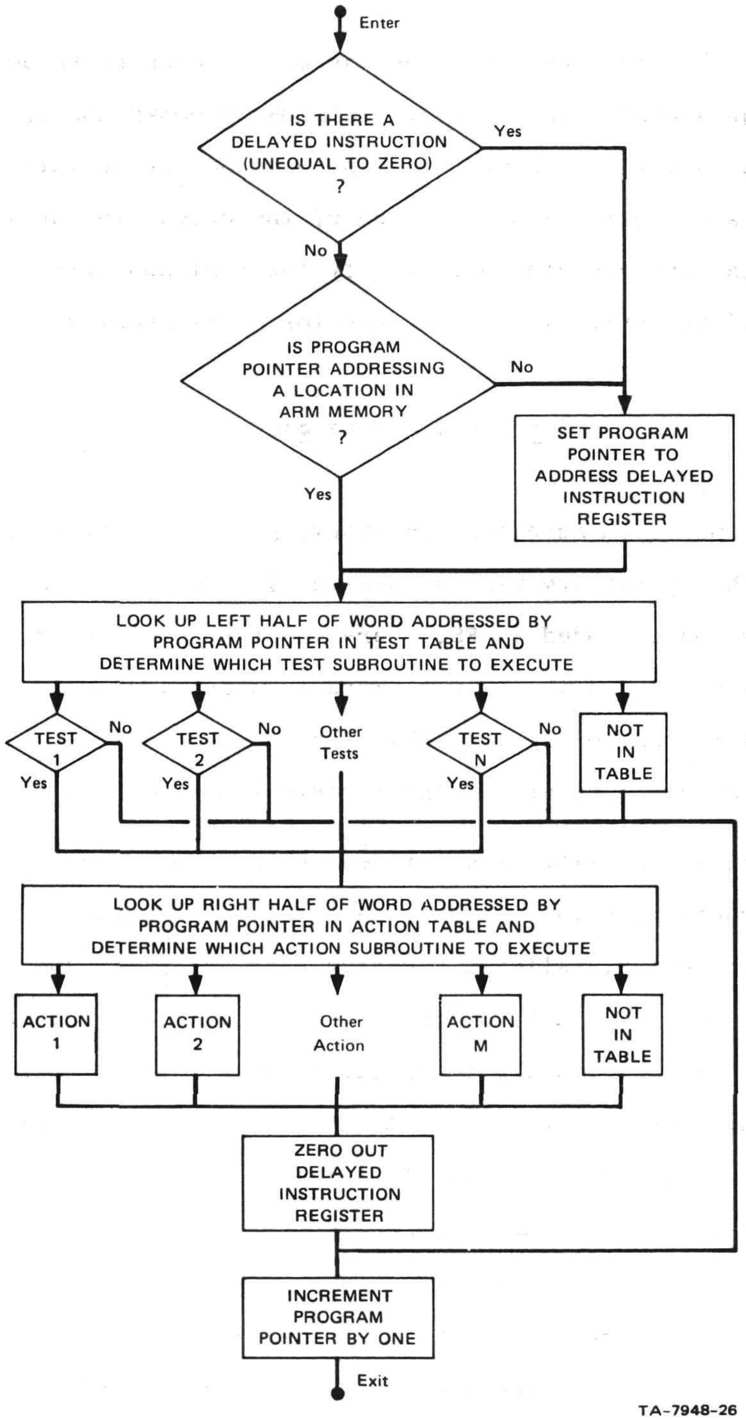


FIGURE 8 INSTRUCTION PROCESSOR. This subroutine carries out single instructions and programs of instructions. It can be seen that, if instructions are being taken from a program in arm memory, an instruction sent from the control station will cause the program to be stopped and the instruction to be carried out.

One advantage of this system is the ability to converse with the arm in a language more natural than the machine language normally used to program small computers. Another advantage in the time-delay situation is that the entire set of subroutines of machine-language instructions for the given test or action need not be transmitted to the remote computer. These subroutines are already built into the remote computer. Only a single instruction need be transmitted.

These instructions also allow short sequences of operations to be sent from the control station at one time, instead of their having to be sent one by one with a wait for a return message after each. Thus, the sequence of commands:

IF FINGERTIP (sensor) THEN STOP

OPEN (jaws)

allows a particular job to be done with one cycle of transmissions through the time delay that would ordinarily take three cycles. Additionally, with long time delays, this sequence of commands specifies a task that would require great caution if performed completely under manual control. In such a time-delay situation, it is difficult to touch an object without producing some overshoot that may knock the object away or without having to use a move-and-wait strategy with successive motions of decreasing amplitude that require considerable time.

Consider, once again, the task described in the manual control section. Let us further complicate the task by assuming that the task is to pick up many pegs and deposit them in the same round hole. In a transmission delay environment, this would be an extremely difficult, time-consuming, and fatiguing task. If we use the decision-response mode of control, one of the many possible sequences of commands might appear as:

```
>BRACE:ABSOLUTE
>DECISION-RESPONSE MODE
@DEFINE BIN

@IF GRAB THEN MOVE TO BIN

@EXIT-DONE

@QUIT
>KNOBS:RELATIVE
>BRACE:ABSOLUTE
>DECISION-RESPONSE MODE
@IF GRAB THEN MOVE TO BIN

@EXIT-DONE
          .
          .
          .
         etc.
```

In the above sequence, the operator initially establishes absolute arm control via the Rancho brace. He then enters the decision-response mode of control and manually moves the arm over the box in which the pegs are to be inserted. Once the arm is in a satisfactory position, he saves the joint angles via the DEFINE command; thus, he may return to that position at any time by merely commanding MOVE TO BIN. At this point, the operator is ready to begin his task. He sets up the reflexive action by using the GRAB test and the MOVE TO response. The operator then manually moves the arm to the vicinity of a peg and attempts to pick it up. Once he has grasped the peg between the tongs of the end effector, the grab test is passed, and the arm automatically switches to computer control and moves to the bin. Once the arm has reached the bin, the computer responds with a done message and the operator quits the decision-response mode. He then transfers control to the potentiometers in the relative mode, and he proceeds to insert the peg in the hole. Having inserted the peg, the operator returns to absolute control from the brace, and the same sequence is once again initiated to pick up the second peg. This

sequence may be repeated until all pegs have been picked up; however, the operator need not redefine the position of the bin each time.

The decision-response mode is clearly advantageous to the operator. Much of the time spent manually moving the arm may be avoided by establishing simple reflex actions based on sensory information. The operator thus assumes more of a supervisory role in remote manipulation tasks, and much of the fatigue and waiting are considerably reduced, with a concurrent rise in the time available to the operator to plan strategy.

C. Algorithmic Remote Manipulation (ARM) Language

If requests for the automatic operations described in the preceding section are taken from a list, the list can be considered a program of motions (an algorithm) to carry out a manipulation task. These programs can provide such simple features as position memory or path memory, or they can perform such complicated automatic tasks as unscrewing a nut from a bolt. A single command specifying that successive commands be taken from a list of commands can be a very powerful and flexible method for producing computer-assisted manipulations.

The effective utilization of such a program, however, requires a means for writing it in an easy-to-use language and a means for assembling (or generating) a list of arm operations from the statements in the language. Under the constraint of a small computer system, we simultaneously developed the separate concepts of the ARM language, composed of the assembler and the instruction set* for the automaton controller previously described. In addition, for program control the following simulated features have been added to the computer controller:

---

*Described fully in Appendix H.

- The 256-word memory, which can be loaded with a list of instructions by a single command from the remote-control station.
- The program pointer, containing the address of (or pointing to) the next instruction to be executed. The address may be that of the delayed instruction register or any of the instructions in the 256-word memory. There are instructions that back up, skip, or specifically set the program pointer.

ARM is an extension of the MHI or THI language developed by Ernst (1961) and of MANTRAN developed by Barber (1967), in that manual inputs from the operator can be used in addition to teletype inputs. Thus, the operator can move his control brace and request that the arm move to "this" position or move "this" joint "this" much, where "this" is a manually specified quantity that is difficult to verbalize, much less to quantify as a joint vector for typewriter input.

The task of collecting objects from a table and depositing them in a bin, as illustrated in Figure 9, can be used to demonstrate a simple ARM program. This is the task of the automatic control experiments described later in Appendix G. A flow chart analysis for the pickup

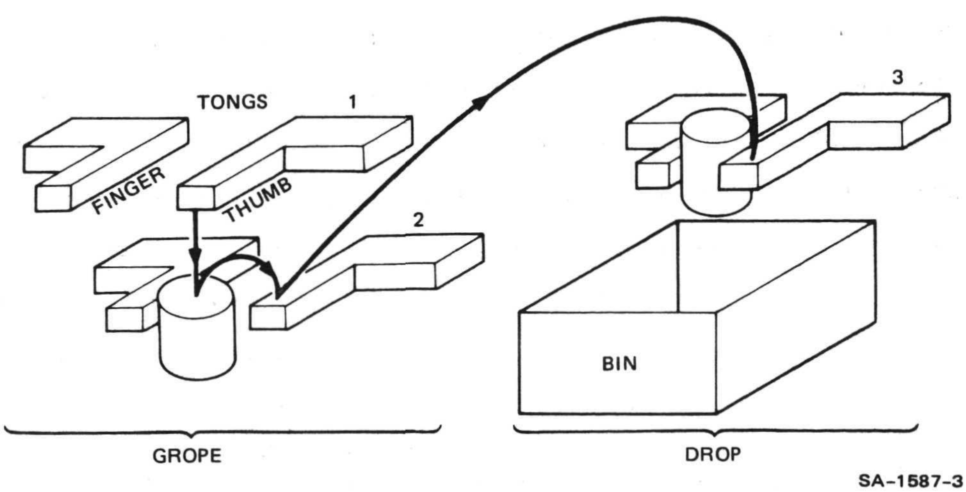


FIGURE 9 MOTIONS OF GROPE AND DROP PROGRAMS

portion of this task is given in Figure 10. An example of a program called GROPE (written in ARM), an algorithm for the actions of picking up an object from a table based on touch information, is given in Table 1.

The entire program is given to an assembler for conversion to a list of numbers (instructions) for execution by the remote computer. Compiling is quite straightforward: Values for the various symbols on each line are simply added together to form the instruction.

Even after the GROPE program of Table 1 has been loaded into the controller's memory and started, the grasping sequence is not begun until the bottom of the index finger or thumb of the end effector has been brought into contact with an object. After the sequence of moves has been finished, the program returns smoothly to the manual control mode by the TRANSFER command. If for any reason the operator wishes to stop in mid-task, he need only transmit any one instruction to the controller.

As can be seen from the GROPE program, the language is quite simple and powerful, needing only 56 12-bit instructions and 3 command vectors for carrying out the pickup. Another ARM program uses 60 instructions and 42 storage locations to direct the manipulator to unscrew a nut from a bolt and deposit it in a receptacle (Hill and Bliss, 1971). Still other ARM programs are being written to center an object in the jaws and yield to an external force. The lengths of many of these programs could have been reduced if inclusion of special procedures had not been necessary to compensate for imprecision in the manipulator used.

Consider once again the example of picking up many objects and depositing them in the same receptacle. This task may be accomplished by using the GROPE program described above and another program called DROP. The following is one of the possible sequences of commands:

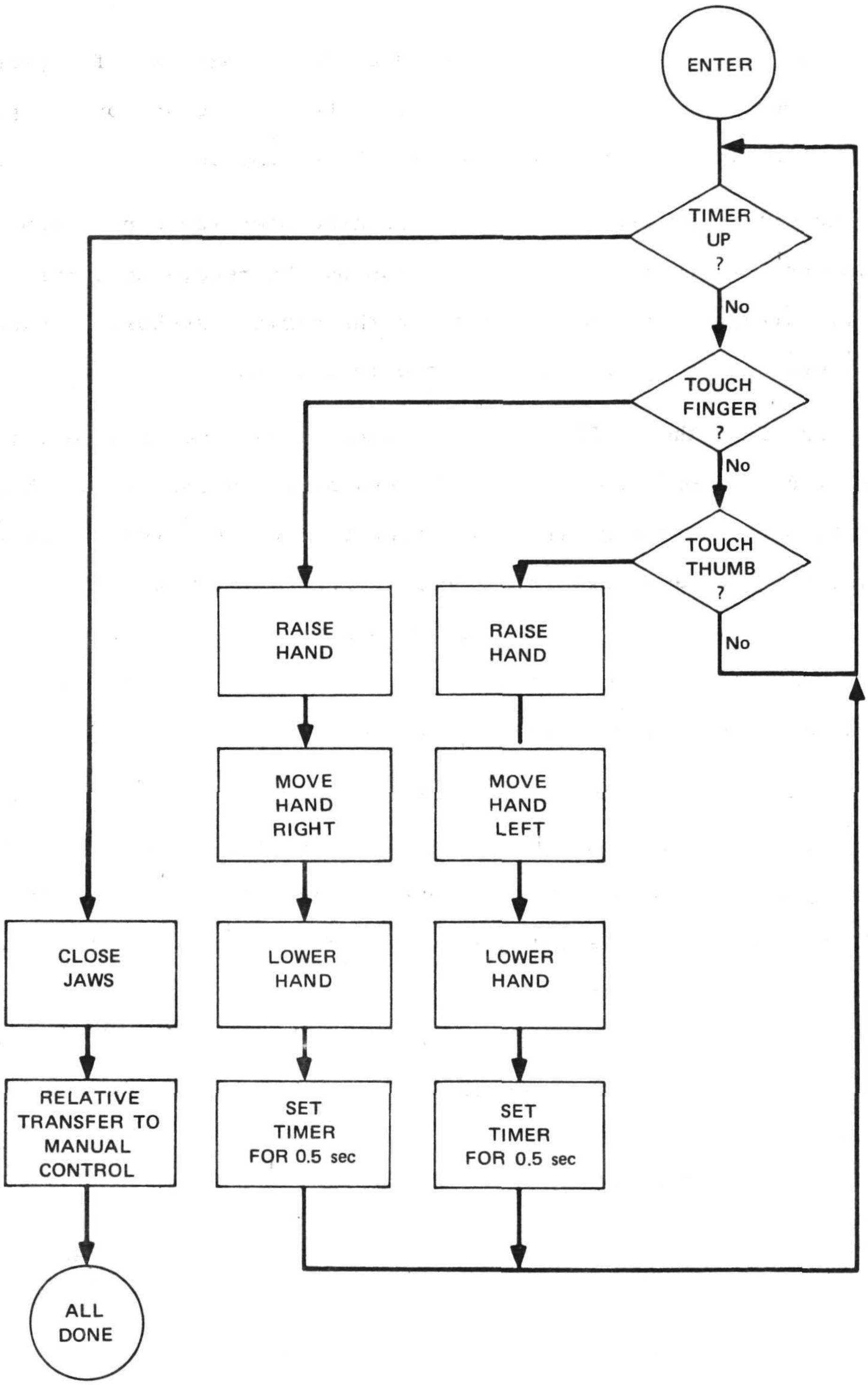


FIGURE 10 SIMPLIFIED FLOW CHART FOR GROPE PROGRAM

Table 1

GROPE: A PROGRAM TO PICK UP AN OBJECT AUTOMATICALLY

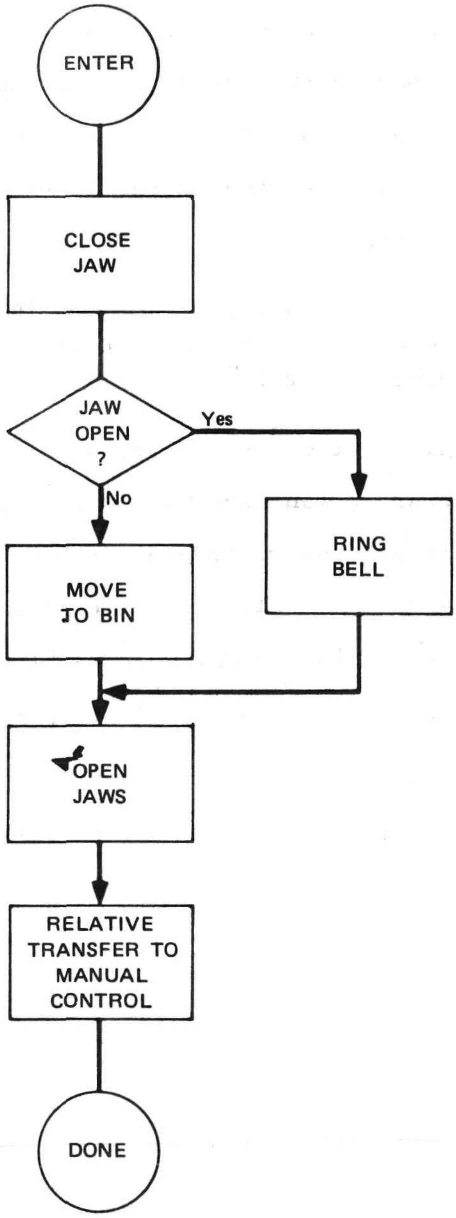
```
BEGIN

        SET THRESH;10
        SET RMASK;7777                          \TEST 100 PCT OF PADS

START:  IF FINGER ON THEN GO TO;RIGHT           \CHECK SENSORS
        IF THUMB ON THEN GO TO;LEFT
        IF WAIT THEN GO TO;START
        GO TO;GOTIT

RIGHT:  DECREMENT BY;ELBO                       \GO UP
        SET CLOCK TO;10 JIFFYS
        IF WAIT THEN REPEAT
        INCREMENT BY;SHOLDR                     \MOVE RIGHT
        SET CLOCK TO;20 JIFFYS
        IF WAIT THEN REPEAT
        INCREMENT BY;ELBO                       \GO DOWN
        SET CLOCK TO;20 JIFFYS
        GO TO;START

LEFT:   DECREMENT BY;ELBO                       \GO UP
        SET CLOCK TO;10 JIFFYS
        IF WAIT THEN REPEAT
        DECREMENT BY;SHOLDR                     \MOVE LEFT
        SET CLOCK TO;20 JIFFYS
        IF WAIT THEN REPEAT
        INCREMENT BY;ELBO                       \GO DOWN
        SET CLOCK TO;20 JIFFYS
        GO TO;START

GOTIT:  INCREMENT BY;TINYELBO                   \GO DOWN TINY BIT MORE
        CLOSE JAWS                              \GRAB IT
        SET CLOCK TO;100 JIFFYS
        IF WAIT THEN REPEAT

        SET RMASK TO;7777                       \GRAB ANYTHING?
        SET THRESH TO;60
        IF RTEST THEN GO TO;START-4

        TRANSFER                                \TRANSFER SMOOTHLY TO
                                                \MANUAL CONTROL

ELBO:           0;0;0;20;0;0;0
SHOLDR:         10;0;0;0;0;0;0
TINYELBO:       0;0;0;5;0;0;0

END
```

```
>BRACE:ABSOLUTE

>COMMANDS FROM*GROPE

>COMMANDS FROM*DROP

>COMMANDS FROM*GROPE

>COMMANDS FROM*DROP
        .
        .
        .
       etc.
```

In the above sequence, the operator establishes initial absolute control from the brace as before. He then specifies that the ARM program called GROPE be run. He then moves the arm manually until either the finger or thumb bottoms contact the object. After the automatic pickup, control is transferred back to the operator. He then issues commands for the program called DROP to be executed. The logic for DROP is illustrated in Figure 11. DROP checks to make sure the object is still there, and then commands the arm to move to the predefined position of the bin and to open the jaws. DROP then automatically returns absolute brace control to the operator. The operator then needs only to repeat the above sequence until all of the blocks have been collected. Thus, once again, the operator is relieved of much of the burden of direct control and is able to spend more time in a supervisory role. Additionally, as the level of control becomes more automatic, the number of commands the operator must issue is reduced, making the task less difficult.

D. Dynamic ARM Programmer (DYNARM)

Two features that an automatic arm controller should possess are not afforded by the ARM language. First, there should exist the capacity for on-line programming, thus obviating the need for separate program



SA-1587-5

FIGURE 11 SIMPLIFIED FLOW CHART FOR DROP PROGRAM

writing, assembly, and debugging. Second, there should be a means for specifying alternative actions, so that the failure of a task does not immediately burden the operator before various other reasonable alternatives have been tried. For example, the operator might desire to use the previously described GROPE program to pick up objects of significantly smaller size than anticipated. This situation might then require that a different sensor threshold be used in the GROPE program. He would thus like to say something like

> "Run the GROPE program, and if it fails, decrease the sensor threshold and try again. If it still fails, then return to manual control."

To implement this desire through typed instructions to a computer, the use of list processing language was investigated. The results of this study suggest that a future language based on this mode of communication can be both simple and flexible. To illustrate these ideas, the structure of a higher level language is outlined, together with some examples of how it can be used to specify particular tasks.

Since an ARM program is merely a sequential list of words that are interpreted as instructions, an on-line list processor would be a more sophisticated mode of control. There are eight basic types of list operations (Katzan, 1970):

- Reference the ith element to examine or modify it in some way.
- Delete the ith element.
- Insert a new element just before or after the ith element.
- Combine two lists.
- Divide a list into two other lists.
- Copy a list.
- Sort the elements of a list by using a specific value within an element.

- Search for occurrence of an element with a set of desired properties.

If these operations were available, the operator could, at run time, combine ARM programs (lists), change constants, delete certain decision-response pairs, and insert new instructions tailored specifically to the task at hand. In DYNARM there will be at least four functional expressions, and one run statement, which may be conditional or unconditional. These expressions are described in Table 2.

If we return now to the collection task previously described, the utility of DYNARM can be easily seen. To pick up an object and drop it into a box, the operator could enter the DYNARM mode and type:

```
SYNTHESIZE MOVE (GROPE,DROP)

RUN MOVE
```

Using list processing, the DYNARM processor would then combine the two ARM programs called GROPE and DROP into a single program called MOVE and transfer control to the decision-response mode where MOVE would be executed. Suppose, however, that the operator desired to pick up objects that were significantly smaller in size than he had anticipated. He might then like to change the sensor threshold from 10 to 5. The sequence would be:

```
MODIFY (SET THRESH;10,SET THRESH;5,GROPE)

SYNTHESIZE MOVE (GROPE,DROP)

RUN MOVE
```

If a sensor threshold of 5 were still too great, the operator would have to repeat the above sequence until the threshold had been sufficiently

Table 2

ELEMENTS OF THE DYNARM LANGUAGE

<table>
<tr><th>Statement</th><th>Type</th><th>Syntax</th></tr>
<tr><td>DELETE</td><td>Functional</td><td>DELETE (<stmt to be deleted>, <arm pgm where stmt occurs>)</td></tr>
<tr><td>MODIFY</td><td>Functional</td><td>MODIFY (<stmt to be modified>, <new stmt>, <arm pgm where stmt occurs>)</td></tr>
<tr><td>SYNTHESIZE</td><td>Functional</td><td>SYNTHESIZE <new arm pgm name>
(<arm program$_1$ | decision resp pair$_1$>,
<arm program$_2$ | decision resp pair$_2$>,...
<arm program$_n$ | decision resp pair$_N$>)</td></tr>
<tr><td>REVERSE</td><td>Functional</td><td>REVERSE (<arm pgm to be reversed>, <name of resulting arm program>)</td></tr>
<tr><td>RUN</td><td>Unconditional run</td><td>RUN <arm pgm></td></tr>
<tr><td>RUN</td><td>Conditional run</td><td>RUN (<arm pgm | decision> → <arm pgm | response>, <arm pgm> | resp>;...
<arm pgm | decision> → <arm pgm | resp>, <arm pgm | resp>)
↕
RUN (<if$_1$> → <then$_1$>, <else$_1$>;...<if$_N$> → <then$_N$>, <else$_N$>)</td></tr>
</table>

reduced. This problem could be circumvented by using the conditional run statement. The sequence would be:

```
SYNTHESIZE MOVE(GROPE,DROP)

RUN (MOVE→RETURN,DECREMENT THRESH∧REPEAT)
```

The run statement would be interpreted as

> "Run move. If it is successful, return to the operator; otherwise, decrement sensor threshold, and try again."

As a final example of DYNARM, consider the problem of collecting all the objects on the table using only one sequence of commands; the two ARM programs GROPE and DROP; and a program called SEARCH. The sequence would be:

```
SYNTHESIZE MOVE (GROPE,DROP)

RUN (SEARCH→MOVE,DONE;MOVE→REPEAT,RETURN)
```

The run statement would be read as:

> "Run the SEARCH program. If it is successful and an object is found, then move it to the box. Otherwise, if no more objects were found, then return control to the operator as done. Now, if the object was moved to the box correctly, then repeat the operation. If not, then return to the operator for assistance."

The use of DYNARM will introduce considerable flexibility into the automatic arm responses. This, together with the fact that even now one ARM program is capable of calling another, indicates that from only a few small ARM programs performing simple tasks, a wide range of complicated tasks can be performed on-line. Additionally, the operator will be able to specify alternatives and actions. When DYNARM is implemented,

the question will arise: What constitutes success in a task? In the simple decision-response tests and actions, success is clear-cut; a sensor is either on or off. In more complicated tasks, such questions arise as "Did I grab it correctly?" "If not, is it still acceptable?" These questions are ill-defined and will pose interesting problems.

E. The Planner

The planner currently exists only as a somewhat futuristic concept, and to think it can soon be implemented is being optimistic because it requires a model of the human operator specifying certain remote manipulation quantities and relationships that have not yet been determined.

Briefly, the planner is a mode of control that is interactive and that will understand English language descriptions of various tasks. The planner will then break these tasks into smaller ones, and the smaller ones into still smaller ones, until it knows how to perform these small subtasks. Considering the operator involved, the various control modes available, and the allowable risk of failure given, the planner should then decide who (man or computer) should carry out the subtasks and under which control mode they should be performed. Notice that the sequence of subtasks is not considered; it is implicit in the process by which subtasks are derived. Finally, the planner should generate DYNARM types of commands to accomplish the task.

It has been felt that the existence of this concept has been useful, as it provides a framework to which various aspects of our work may be tied for eventual integration into a single system. The concept of the planner has raised such questions as: "How is a task described?" "How is a task further subdivided into subtasks?" "Which subtasks are common to most types of manipulations?" "How do we measure performance?"

The planner is mentioned here only to demonstrate its proposed relationship to the other control modes and to stimulate thinking concerning the various aspects of remote manipulation.

F. Summary

The various existing and planned control modes are illustrated in Figure 12, which shows the relationship of each mode to the others. It also illustrates how the human operator may exercise any of the control modes and at the same time exercise all lower level modes.

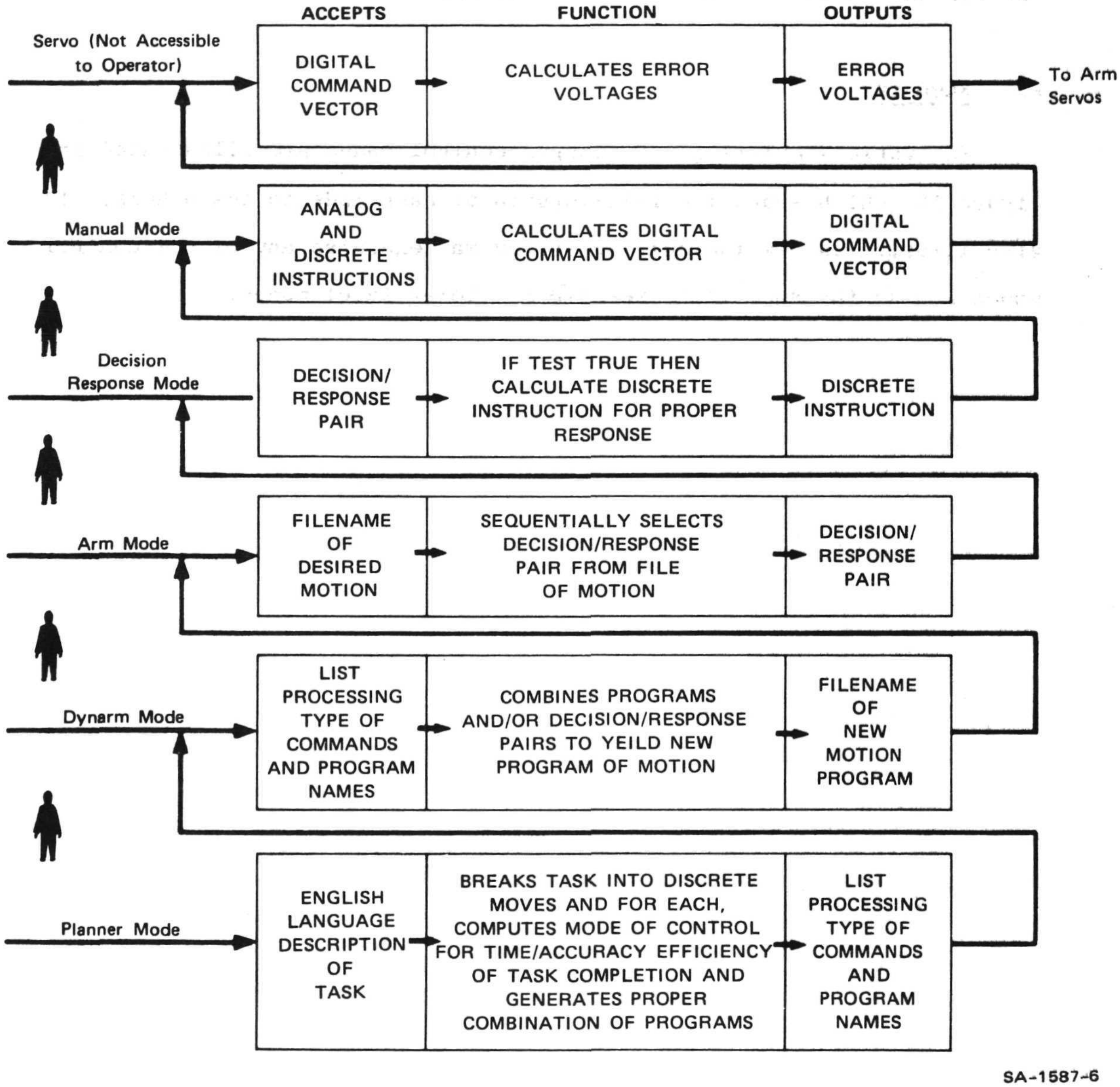


FIGURE 12 HIERARCHY OF MULTIMODED ARM CONTROL SYSTEM

IV TRANSMISSION DELAY

In communication systems with a time delay, such as those used in exploration of the moon or the planets, direct control by a human operator becomes a very slow and laborious process. The problem is that the human operator cannot see the results of an action until some later time determined by the time delay. During this period of delay, the environment may have changed or an action may have been too extreme. The operator is thus forced into a move-and-wait situation in which his moves are cautious while he waits for feedback on the results of his actions. Physical fatigue and frustration may compound the problem.

A. Move-and-Wait Strategy

A time history of master moves and the subsequent slave moves is illustrated in Figure 13. A move is defined as the period of time between the beginning of a master move and the beginning of the subsequent master move. Each master move is considered to occur in three phases (Sheridan and Ferrell, 1963): move time, wait time, and reaction time, as defined below.

- M_m--Duration of master move.
- M_w--Time from end of master move to end of slave move.
- M_r--Time while master reacts to the consequences of his move and decides upon a subsequent move.

When a simple move-and-wait strategy is being used, the total task time can be expressed in terms of these times using the following formula:

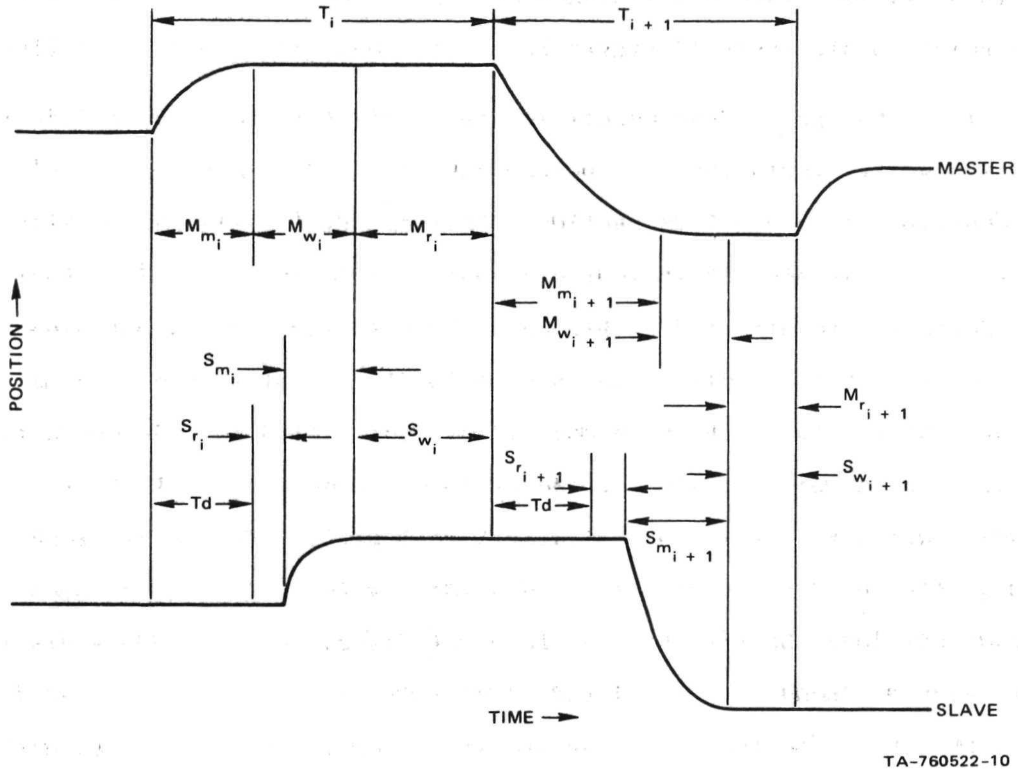


FIGURE 13 TIME HISTORY OF THE i^{th} AND THE $i + 1^{st}$ MOVES FOR A MOVE-AND-WAIT SITUATION

$$\text{Task Time} = \sum_{i=1}^{N} \left(M_{mi} + M_{wi} + M_{ri}\right) \quad ,$$

where N is the total number of moves required to complete the task.

A complete description of the situation, however, requires the specification of both the system transmission delay and the slave movement times defined below that correspond to the previous master move times.

- T_d--Round trip transmission delay
- S_r--Slave reaction time
- S_m--Duration of slave move
- S_w--Same as M_r.

If the master follows a true move-and-wait strategy and does not move again until the slave has finished moving (simple move-and-wait strategy), the relationship shown in Figure 13 exists among the above quantities.

To investigate these quantities and their relationship, a transmission delay simulation was incorporated into the arm control system, as described in a previous section. Preliminary investigations with delays up to five seconds indicated a considerable deviation from Sheridan and Ferrell's result; the simple move-and-wait strategy is not used. The longer the time delay, the more frequently complex moves are made before the result of the main move is evident. With a five-second time delay, for example, frequently two or three moves are given before their results are seen, as if the operator were impatient to see his results. Apparently, he has the urge to make a move every 1-1/2 to 2 seconds, no matter how long the time delay. In other cases, he overreaches his target and makes a second move while the first move is in progress. Examples of both simple and complex moves are indicated in the chart recording of Figure 14 obtained with the chart recorder monitor described in Appendix D.

B. Performance Monitor

A performance monitor was created and implemented to study (1) the complex move-and-wait strategy and (2) the movement and waiting times with different transmission delays, with different visual and tactile feedback, and with different arms. The purpose of the performance monitor is to measure and tabulate the movement and waiting times indicated in Figure 13 with considerably greater accuracy and reliability than a human can with a stopwatch.

The performance monitor consists of two parts: an on-line program for move detection and an off-line program for numerical analysis. The on-line program detects the beginning and end of moves by using derivatives

FIGURE 14 RECORD OF MASTER AND SLAVE MOVES WITH THREE-SECOND TIME DELAY. Master moves denoted by arrows are labeled according to slave motion: 1 indicates move while slave stationary (simple move-and-wait), 2 indicates move while slave moving (move-while-moving), and 3 indicates an additional move before result of first move seen (complex move-and-wait). (One time division = 2.5 seconds.)

of the individual joint angles. In total, 14 derivatives (7 master and 7 slave joint angles) are updated and digitally filtered* every 1/30th of a second. If any of the master or slave joints exceeds a predetermined threshold for motion during a 1/30-second period, a note of the fact is made in separate master and slave move detection delay lines. These delay lines (or shift registers) record whether or not a move was detected during 12 successive 1/30-second intervals. From this intermediate data, decisions are made to determine whether a master or slave move has begun or ended. The flow chart of the performance monitor is given in Figure 15. The criteria for detecting the beginnings and ends of moves that have proved successful are defined below.

- Move criterion--Velocity threshold is exceeded during the current 1/30-second interval, and it will be exceeded on 5 of the next 12 intervals.
- Done criterion--Velocity threshold is not exceeded during the current interval, and it will not be exceeded more than once in the next 12 intervals.

Two total task measurements are also obtained. The on-line program counts the number of 1/30-second intervals taken to complete a task and prints the total at the end to permit the calculation of task duration. Additionally, it accumulates the current delivered by the 24-volt servo power supply every 1/30th second, using the current sensor described in Appendix D, and prints the total at the end of the run to permit calculation of the energy consumed.

---

*The 0.25-second exponential filter is obtained by multiplying the previous velocity by 7/8 and adding the new velocity. Limiting these filtered velocities to twice the detection threshold equalizes the delay for detecting moves and waits.

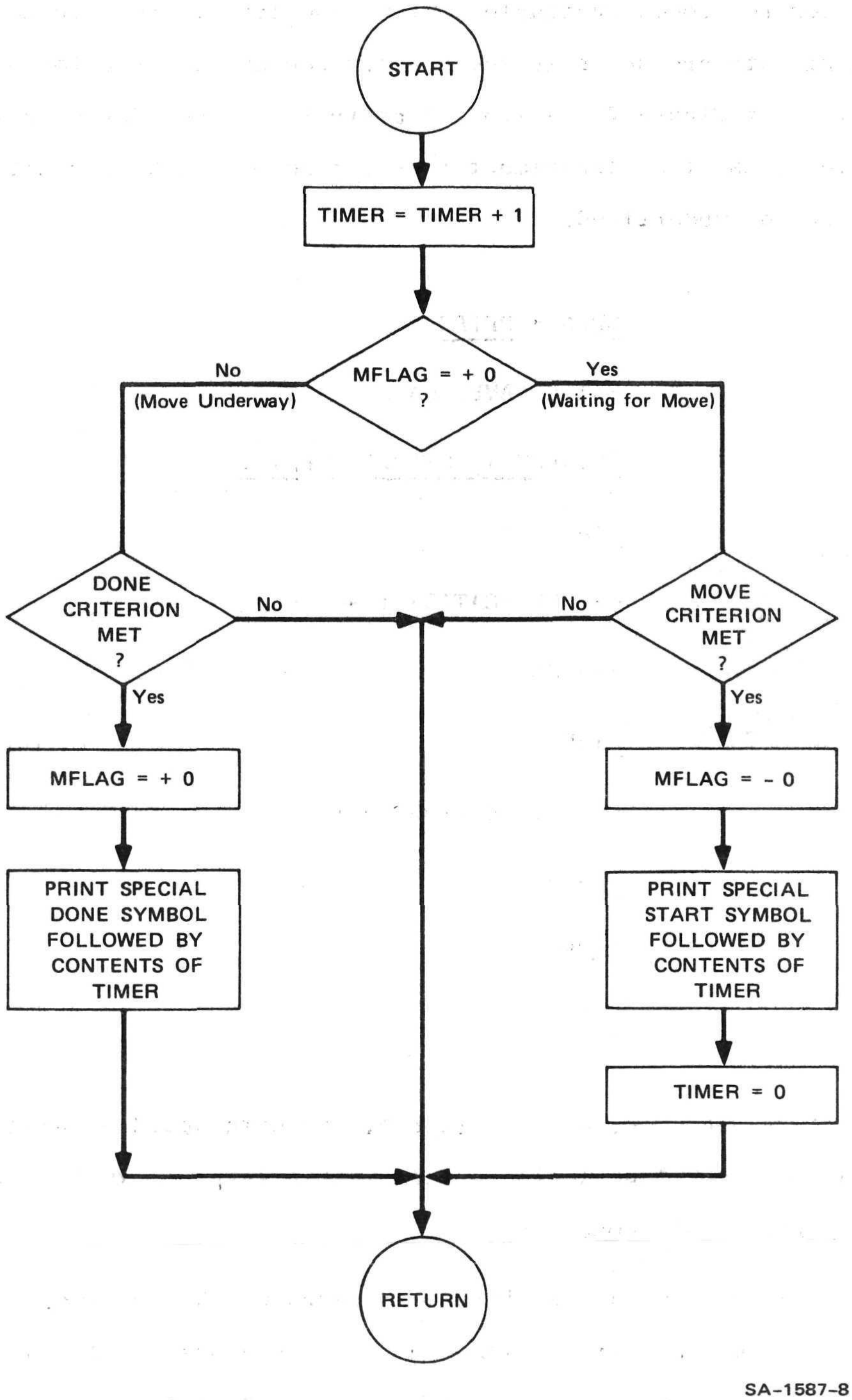


FIGURE 15 ALGORITHM FOR DETECTING THE BEGINNING AND END OF MOVES. Identical detectors monitor master and slave for moves.

The control codes available to the experimenter for accumulating and logging data are shown in the performance monitor section of the control tree of Figure 7. A typical control sequence for carrying out two replications of an experiment is shown below, with information typed by the operator underlined.

```
OPEN * PFILE

ENTER NAME, DATE

SUBJECT 1, 6/10/72 TEST 2

START

---REPLICATION 1---

FINISH

START

---REPLICATION 2---

FINISH

CLOSE

QUIT
```

Following this sequence, a paper tape of the data would be punched off-line, and the paper tape would be read into a larger computer with FORTRAN capability for final processing.

The off-line program tabulates and averages the intermediate times determined by the performance monitor. It has a basic calculation that yields the three quantities defined below for any master or slave move beginning or ending.

- Total number--The number of moves.
- Distribution of times--The distribution of move times is obtained from the times data logged under the on-line program. The actual times are grouped into bins, and a histogram is printed out.
- Mean time--This time is computed from the distribution of move times.

The time to be analyzed by the off-line program is determined by particular start and end points. The same set of symbols indicating master or slave moves or end of moves used by the on-line program are used. The basic calculating package searches through the intermediate data file looking for the particular symbols indicated, and it is thus capable of tabulating any of 20 different manipulation times from the same data file. Files can be analyzed individually or merged to obtain averages across subjects or conditions.

C. Preliminary Analysis

An experiment was run to gather some preliminary qualitative data by using the performance monitor. The experiment consisted of picking up a randomly placed block from a table, using a time delay of 1.7 seconds. The experiment was started when the end effector passed through a plane parallel to and one foot above the table. After the block had been picked up, and the end effector had again passed through the plane, the experiment was terminated. This operation was repeated eight times in succession by both an experienced and an inexperienced operator using the Rancho brace. The inexperienced operator made eight more runs using the potentiometer bank rather than the brace to control the arm.

The data from the performance monitor were arranged into a histogram showing the number of moves and their duration for all eight runs. This is illustrated for both master and slave arms in Figures 16 and 17. From

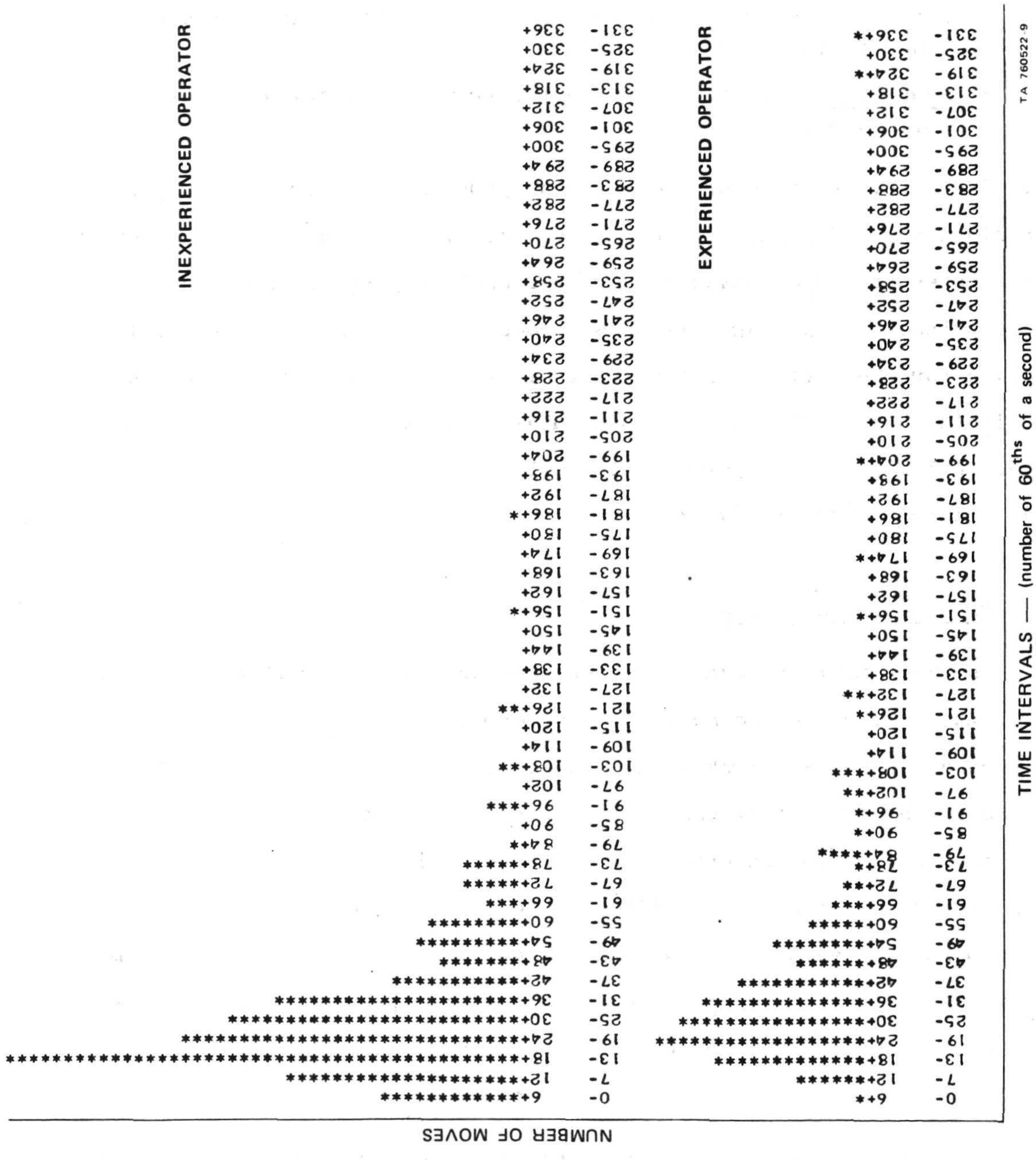


FIGURE 16 DISTRIBUTION OF MASTER MOVE TIMES FOR EIGHT ATTEMPTS

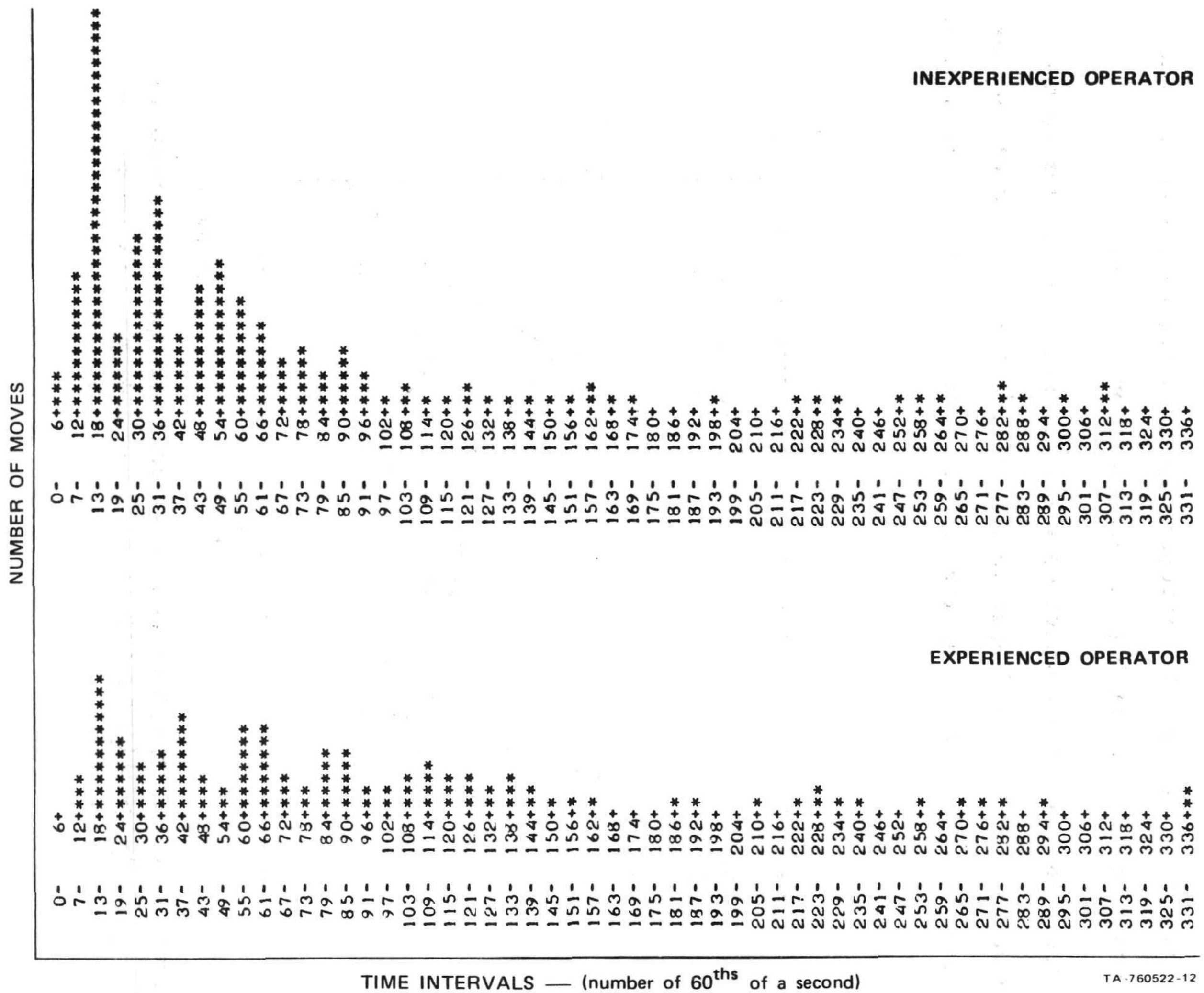


FIGURE 17 DISTRIBUTION OF SLAVE MOVE TIMES FOR EIGHT ATTEMPTS

the histograms, it can be seen that the inexperienced operator made significantly more moves of short duration than did the experienced operator. This indicates a more cautious strategy owing to unfamiliarity with the move-and-wait situation. As expected, the inexperienced slave made a correspondingly greater number of moves of short duration.

Figure 18 shows the completion times for each attempt for both experienced and inexperienced operators. The completion times of the inexperienced operator increased for each attempt, whereas the times of the

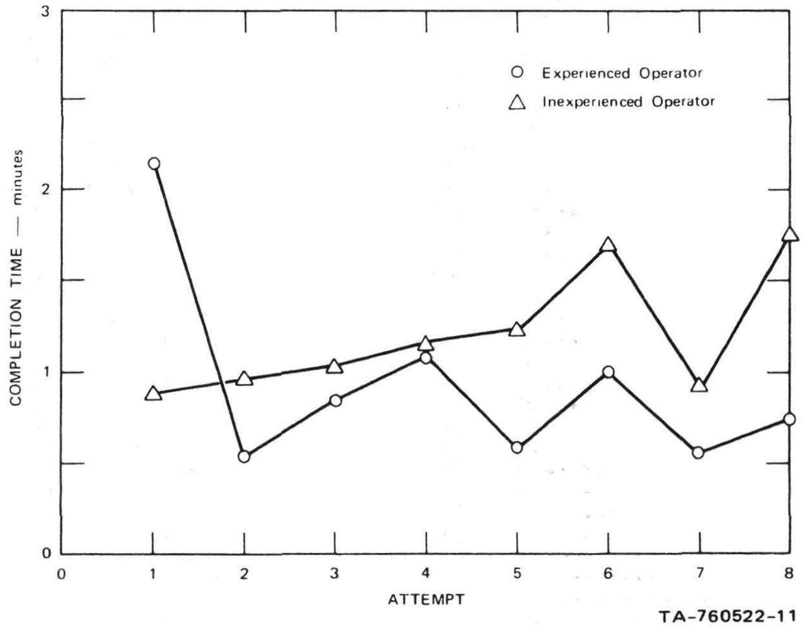


FIGURE 18 COMPLETION TIME VERSUS ATTEMPT FOR THE EXPERIENCED AND INEXPERIENCED OPERATOR. Transmission delay is 1.7 seconds.

experienced operator seem generally to have declined. Both behaviors do not seem surprising. In the case of the inexperienced operator, fatigue became a factor. This probably accounts for the rising trend in completion

times. In fact, the operator complained of noticeable fatigue after the fourth attempt and was allowed to rest after the sixth attempt. In the case of the experienced operator, fatigue did not seem to be a dominant factor, but rather the reacquisition of a previously learned skill. The times seem to have been declining to some previously attained performance level. As was also expected, the completion times for the inexperienced operator were generally higher than those for the experienced operator.

Figure 19 shows task completion time plotted versus the number of moves per task. This figure suggests that an index of performance (I_p) might be defined as the slope ($\Delta Tc/\Delta N$), with a greater slope indicating inferior performance. The presence of such a measure of performance might prove to be extremely useful in evaluating man's adaptability to

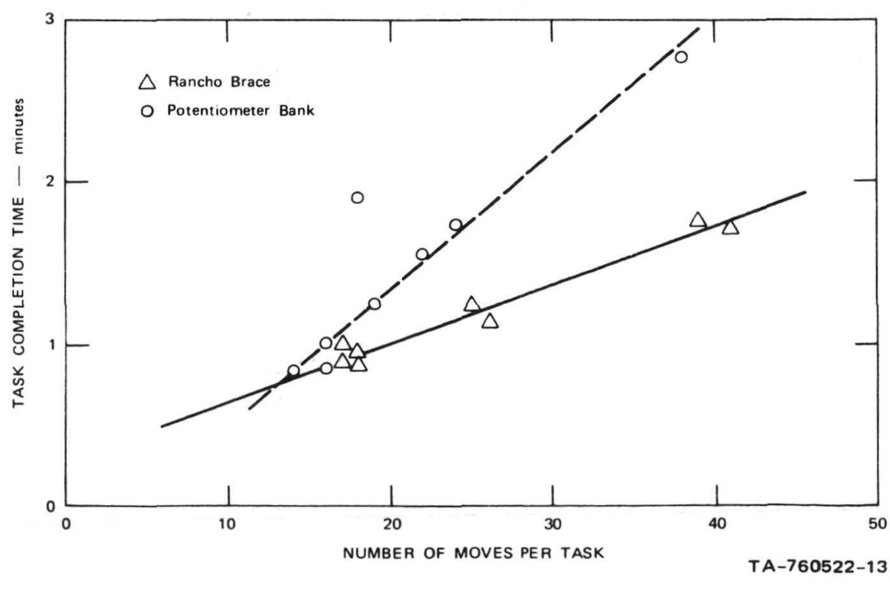


FIGURE 19 TASK COMPLETION TIME VERSUS NUMBER OF MOVES FOR A SINGLE OPERATOR. Transmission delay is 1.7 seconds.

various types of manipulator controls, the performance of different control systems, the performance of different operators, and the effectiveness of different tactile sensors.

The arm control system, together with the performance monitor, automatic routines, and transmission delay simulation, seems to be a useful experimental tool in investigating many of the subtleties of remote manipulation, both with and without transmission delay. Hopefully, it will be possible to gain a better understanding of how we do even the simplest of tasks, and this knowledge can, in turn, be applied to teleoperator technology.

V TOUCH SENSING AND DISPLAY

One way of understanding the contribution of touch sensing to man's manipulative skills is to provide a teleoperator with different kinds of touch feedback from slave to master for teleoperator control. We can determine what skills the new information has provided by carrying out manipulations with the touch feedback first on and then off. After identifying a skill in this way, we also have proof that the same skill can be obtained from an automatic control system utilizing the same tactile information.

A. Sensor Design

Two touch feedback systems for the teleoperator control system were constructed for experimental evaluation. Each system consists of a set of sensors mounted on a mechanical arm and a corresponding set of tactile stimulators mounted on a control brace. All the sensors are constructed of conducting rubber that is deformed to complete an electrical circuit upon physical contact. The construction of most of them depends on etched wiring on printed circuit boards. Individual sensors activate corresponding stimulators in a binary fashion: A stimulator is either full on or full off. Construction details are given in Figure 20, and descriptions follow.

- The whisker sensor shown in Figure 20(a) consists of thin conducting rubber strips pulled through holes in two-sided printed circuit boards. These parallel connected whiskers have high sensitivity (two grams) because of the mechanical advantage of the whiskers.
- The surface sensor shown in Figure 20(b) consists of a conductive rubber sheet held by sponge-rubber pads above a

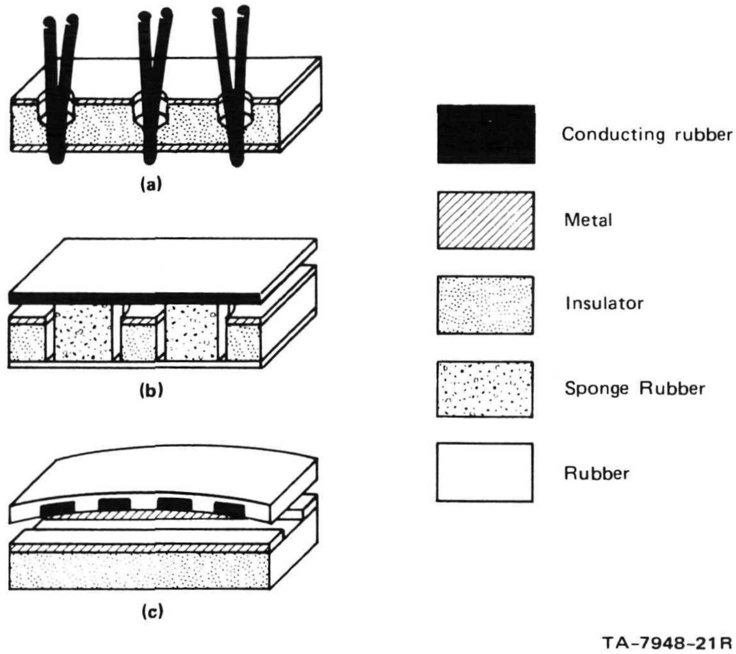


FIGURE 20 SENSOR CONSTRUCTION USING CONDUCTING RUBBER

sheet of single-sided printed circuit board. If islands of copper foil are made by etching the circuit board, then the contact force can be localized to one or more islands.

- The force-distribution sensor shown in Figure 20(c) consists of a sheet of conducting rubber arched over a printed circuit board studded with plated-through eyelets. The shape of the force pattern is measured by measuring the pattern of voltage on the eyelets transmitted from the activated conducting rubber sheet.

B. Hand Contact System

The hand contact system senses and reproduces to the operator the contact between the end effector and the object being touched or manipulated. This system consists of a number of conducting rubber sensors mounted on the outside surfaces of the mechanical hand, as shown in Figure 21 and listed in Table 3.

Table 3

LOCATION OF CONTACT SENSORS ON MECHANICAL HAND

Location	Number	Type*
Tips of tongs	2	(a)
Top of tongs, distal	2	(b)
Top of tongs, proximal	2	(b)
Bottom of tongs, distal	2	(b)
Bottom of tongs, proximal	2	(b)
Back of tongs, distal	2	(b)
Back of tongs, proximal	2	(b)
Web of jaw	1	(a)
Knuckles	2	(b)
Top of wrist	2	(b)
Bottom of wrist	2	(b)

*Refers to Figure 20.

The tongs of the hand are completely covered with these sensors (seven sensors per tong), as are the extreme or protruding parts of the upper hand (seven sensors). The sensors are so arranged that any contact of the hand with a flat surface is sensed, and any contact with the tongs is sensed. Each sensor is connected via amplifying and gating

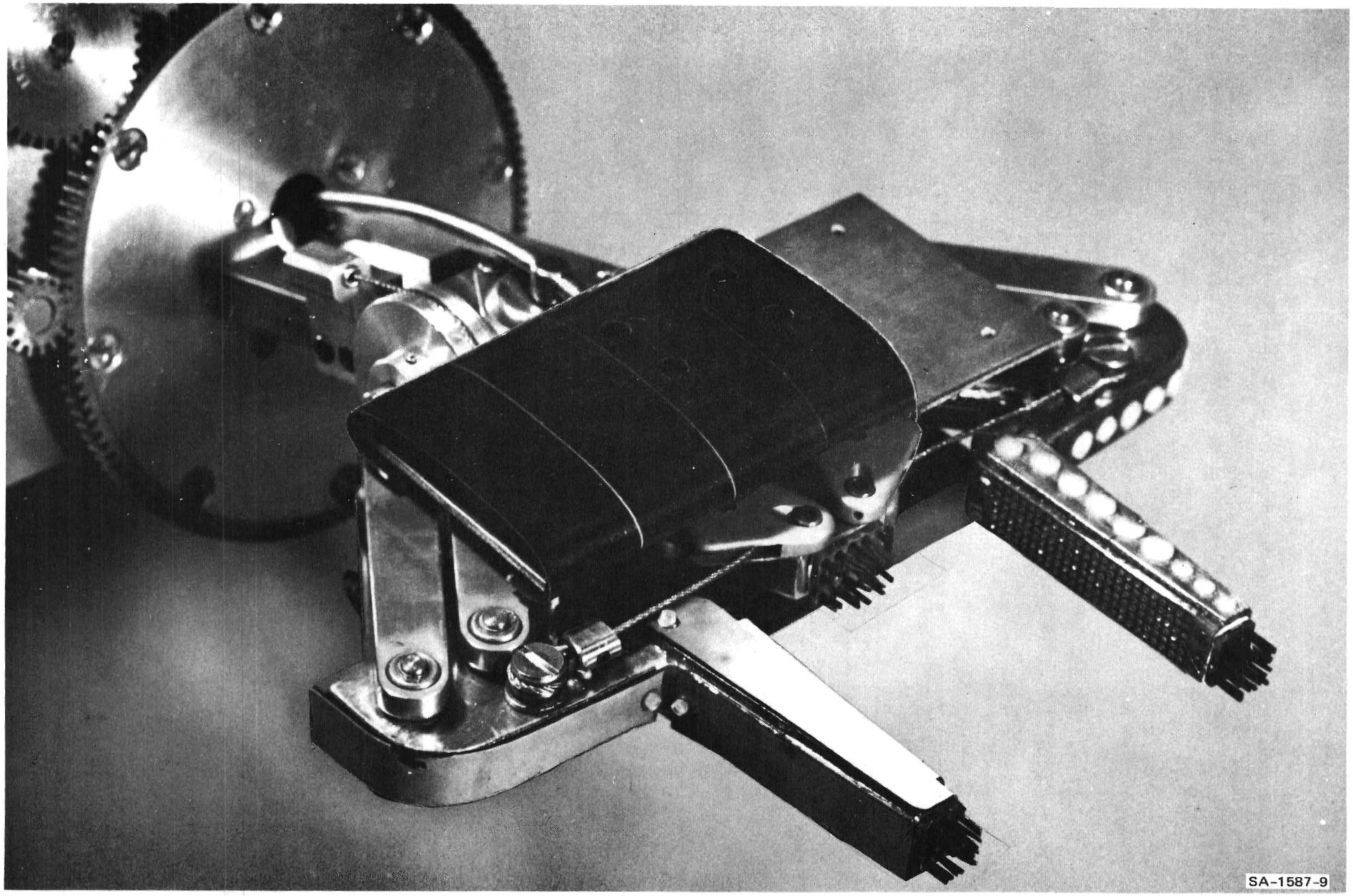
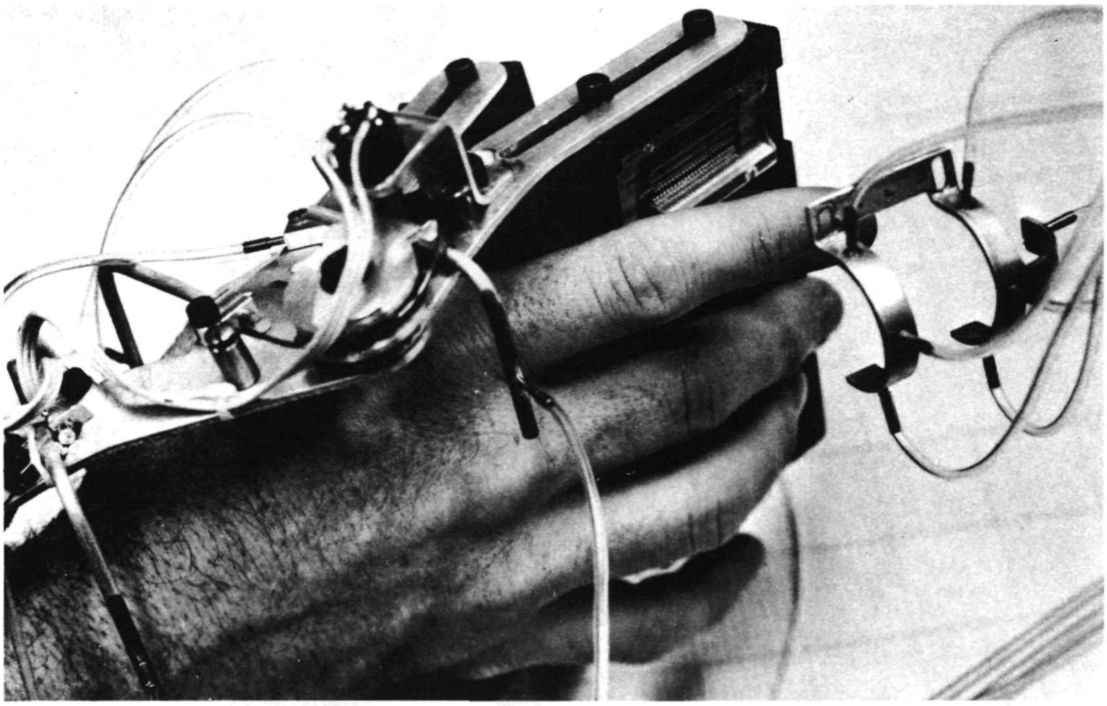


FIGURE 21 COMPOSITE SENSOR CONSTRUCTION. Right half of photograph shows underlying printed circuit boards and matrix of sense pad contacts. Left half shows conducting rubber sheeting in place.

circuits to an air-jet tactile stimulator. The air jets are positioned on the control brace to produce touch sensations on portions of the operator's hand corresponding to the locations of the sensors. Each jet produces an area of pulsating pressure on the skin approximately 3/16 inch in diameter. The arrangement of air-jet stimulators on the control brace is shown in Figures 22 and 23. The construction of the air-jet



TA-760522-5

FIGURE 22 EXPLODED VIEW OF TACTILE STIMULATORS ON THE HAND CONTROLLER. 6 x 24 matrix of piezoelectric stimulators is revealed under operators index finger. An identical unit is under thumb.

stimulators was described by Bliss and Crane (1965). By using the external sensory feedback system, it is possible to (1) reach into a box without the aid of vision and extract a block from it, (2) locate a visually obscured object to be picked up under the tongs and respond by grasping it, and (3) control wrist rotation so that both tongs rest on a flat surface.

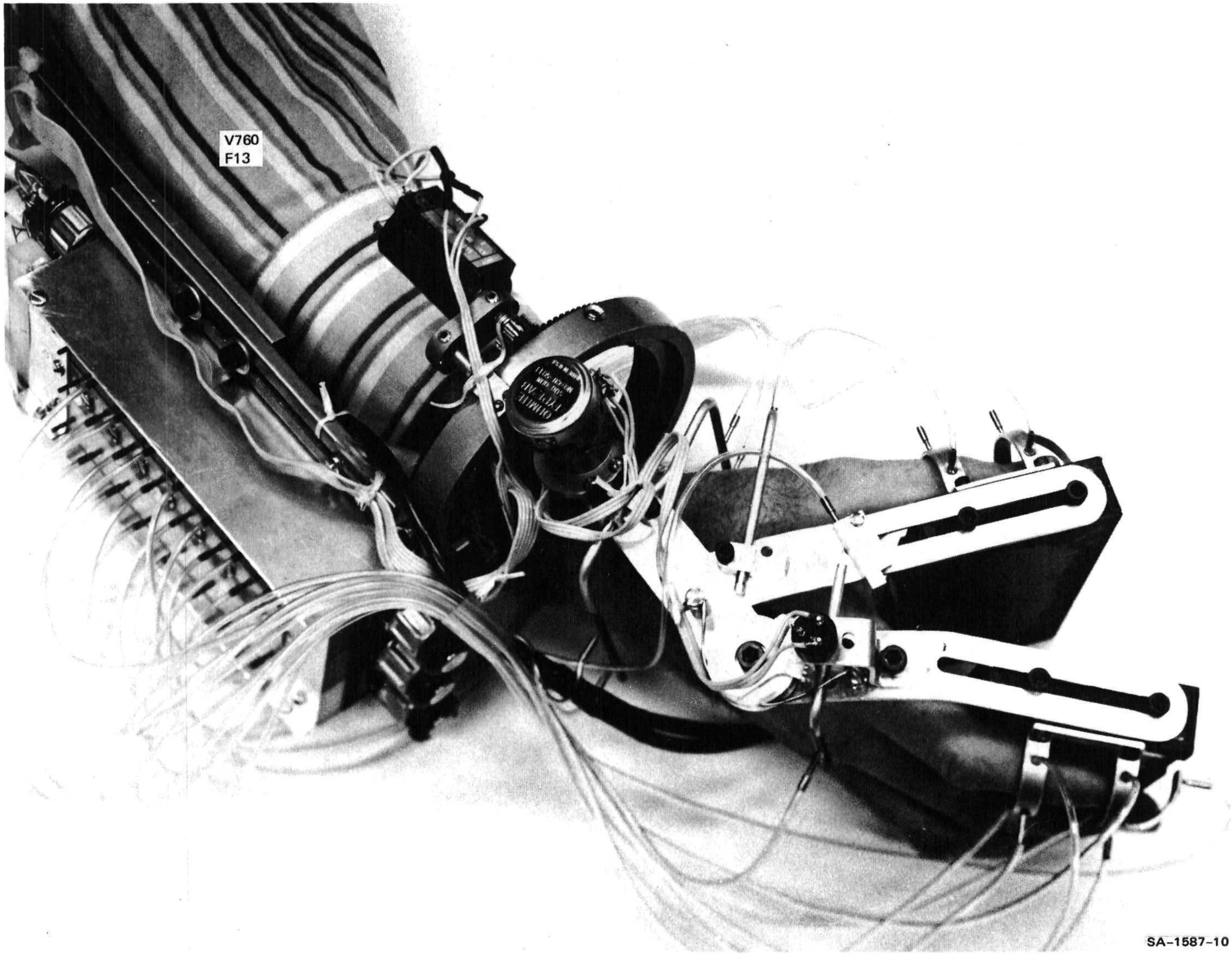


FIGURE 23 AIR-JET STIMULATORS MOUNTED ON CONTROL BRACE. Note attachment of knuckle, webb of jaw and wrist jets in addition to the seven jets on finger and thumb.

C. Jaw Contact System

The jaw contact system senses and reproduces to the operator the shape and location of the object held in the remote jaws. Two sensing pads are built into the tongs of the mechanical hand (as shown in Figure 21). Each of these two opposing pads consists of 144 individual contacts in a 6 X 24 rectangular pattern. Two corresponding 6 X 24 rectangular arrays of bimorphs contacting the index finger and thumb are built into the control brace, as shown in Figure 22. Bimorphs produce a 265-Hz vibration of the skin, restricted to an area about 1 mm in diameter. Thus, the pattern of contact closures is reproduced as a pattern of vibration, enabling the operator to feel on his thumb and index finger the shape and location of the object held in the remote jaws. A complete description of similar bimorph arrays used in shape recognition and reading experiments has been given by Bliss (1969) and Bliss et al. (1970). By using the jaw shape-sensing system, it is possible to (1) pick up an object in the desired part of the tongs, as in the center or at the tip, (2) obtain rotational alignment to a fixed object while gripping it, (3) detect slippage when lifting an object and close the jaws until the slippage has stopped, and (4) pick up an egg without breaking it.

D. Computer-Generated Display

Signals from both sets of sensors are fed into the control computer using the interface described in Appendix B. These signals are used for automatic computer control and visual display of touch information on an oscilloscope screen. The scope display shown in Figure 24 is in perspective, and the motion of the jaws back and forth corresponds to their actual position. Thus, the operator can visually distinguish patterns and points of external contact, as well as amount of jaw closure. Various gripping situations, together with the resulting sensor patterns, are illustrated in Figures 25 and 26.

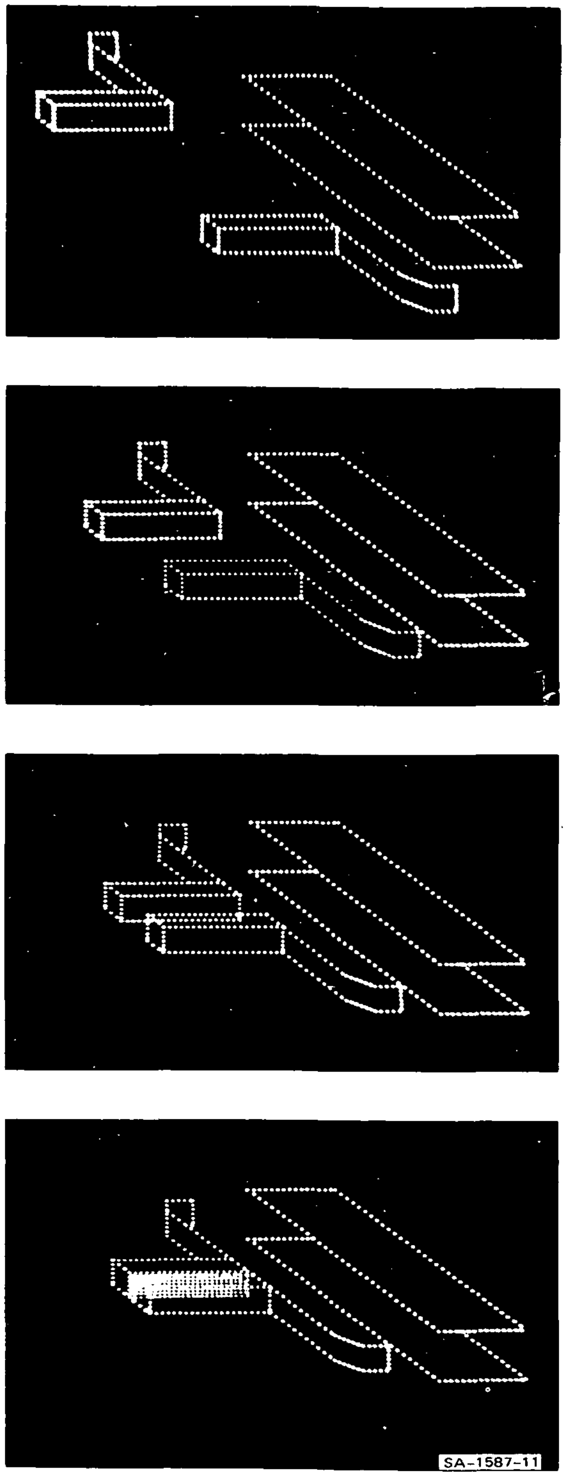
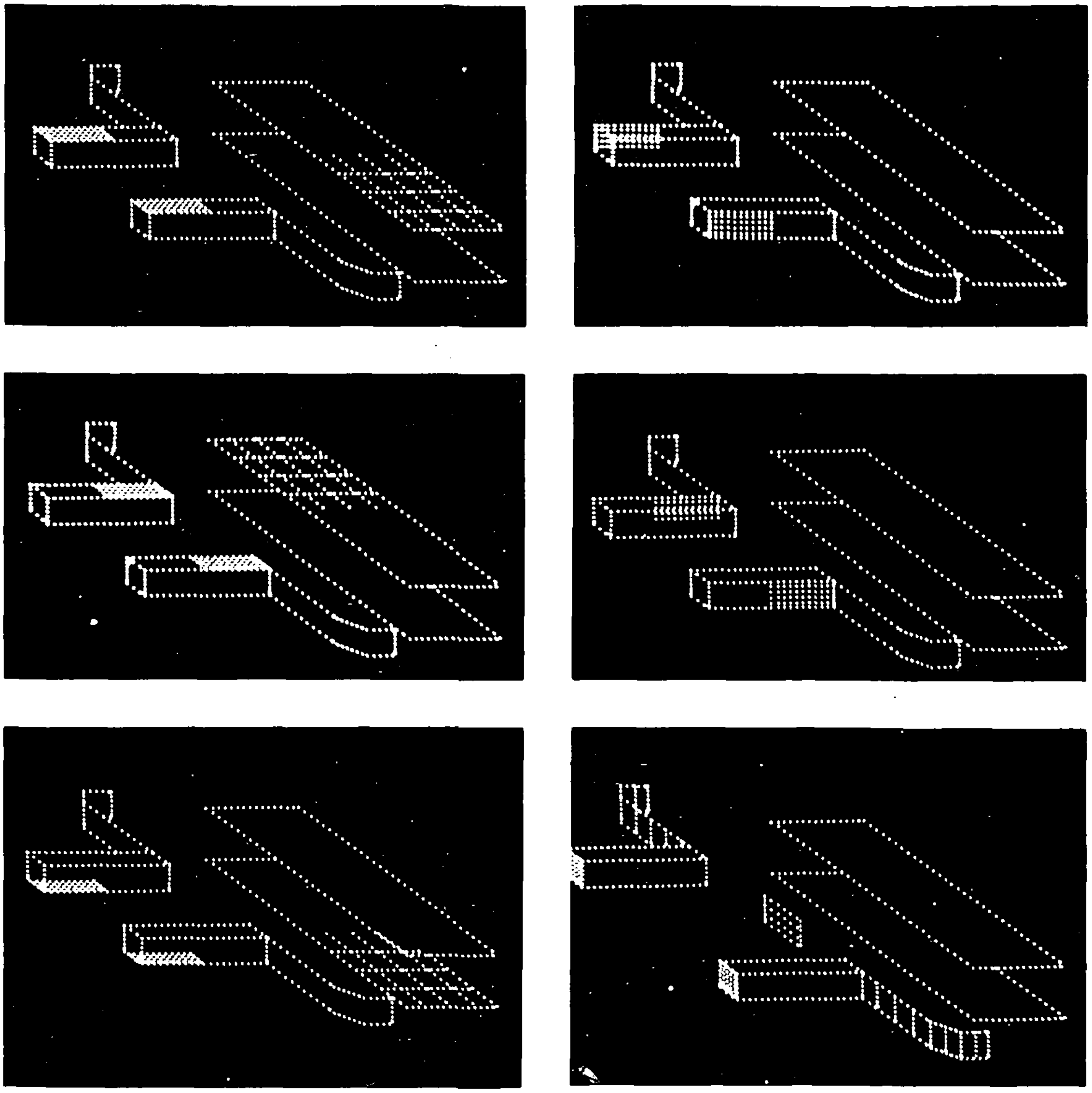
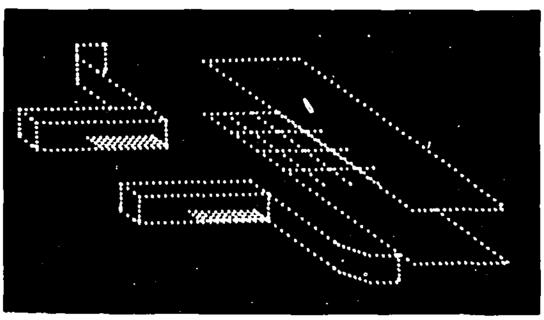


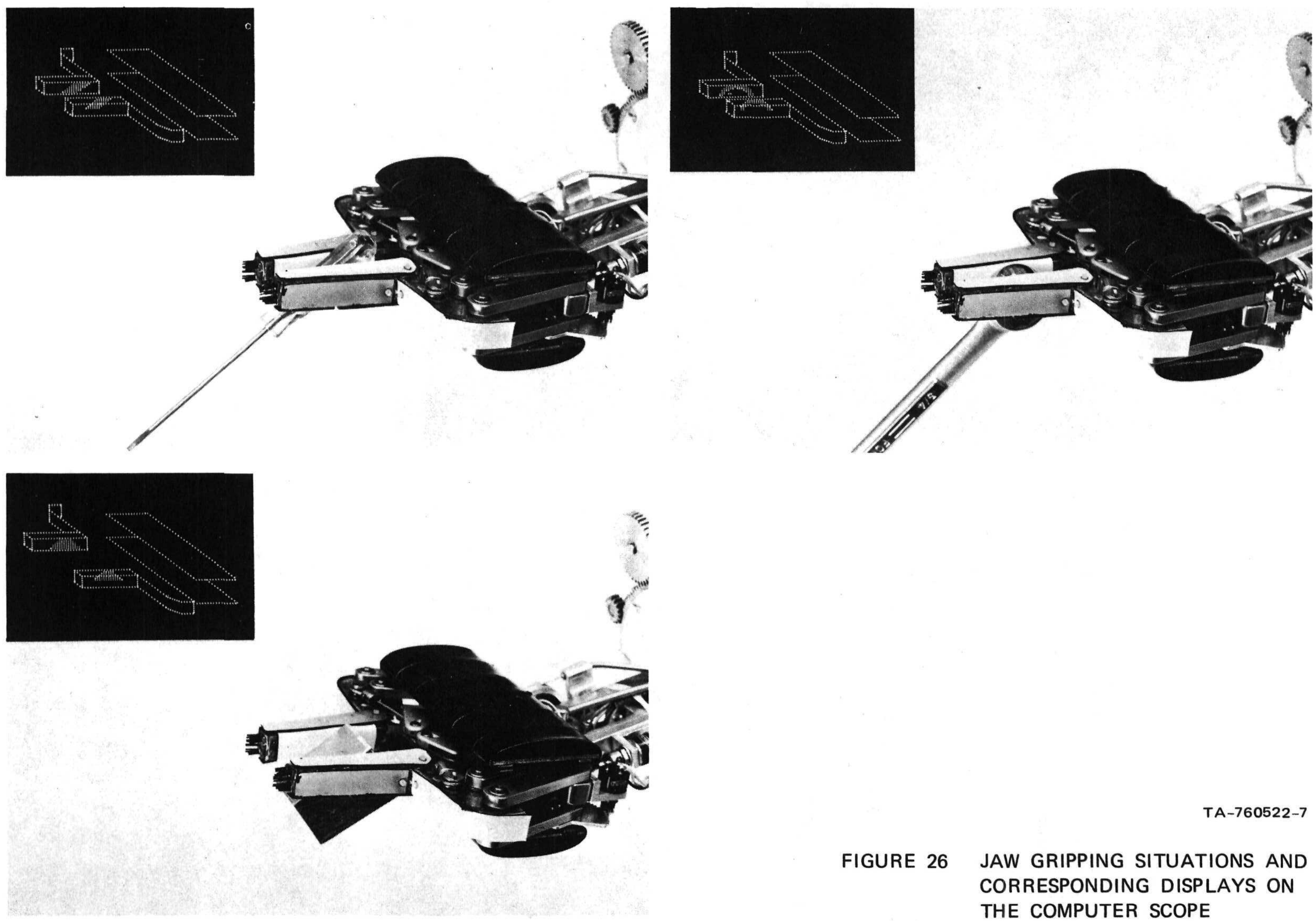
FIGURE 24 DISPLAY SEQUENCE OF JAWS CLOSING





SA-1587-12

FIGURE 25 INDIVIDUAL ACTIVATION OF TACTILE SENSORS AS SEEN ON THE COMPUTER SCOPE



TA-760522-7

FIGURE 26 JAW GRIPPING SITUATIONS AND CORRESPONDING DISPLAYS ON THE COMPUTER SCOPE

VI A PRELIMINARY EVALUATION OF A TELEOPERATOR USING COMPENSATORY TRACKING

A. Introduction

Several aspects of manipulation tasks resemble compensatory tracking. The operator frequently must move the end effector along a particular path, avoid a series of obstacles, capture a moving object, or work from a moving vehicle. The display being viewed provides information on the relative error between the desired object and the position of the end effector. These situations are basically compensatory tracking tasks.

Powerful tools exist for studying compensatory tracking. One is the describing function or linear model of a nonlinear dynamic system of McRuer et al. (1965). Another is the operator's equivalent time delay, a stable and useful performance indicator determined by Jex, McDonnell, and Phatak (1966).

At first glance, the "critical" task of Jex et al. could be used to characterize a man-arm system. By having the man manipulate a joystick with a mechanical arm, one could measure his equivalent time delay, t_e. The procedure, however, only applies to human control of an integrating vehicle, and the inclusion of a particular arm "vehicle" in series with the operator and the integrating vehicle of the task complicates the situation and renders the Jex et al. algorithmic computation of t_e invalid for teleoperators.

A more general approach to the problem is that of measuring the entire operator-arm describing function in a compensatory tracking task. From this data, the equivalent operator-arm time delay can be correctly

determined. The following preliminary experiment was carried out to obtain teleoperator describing function and remnant data to see how they could be used for comparative evaluations of manipulator systems.

B. Apparatus

The compensatory tracking task was carried out by a small digital computer (LINC-8 with 8k of memory). The computer generated the sum-of-sines command signal and performed the Fourier analysis of the error and response signals on-line. The block diagram of the control situation is shown in Figure 27. Although the computer program is capable of carrying

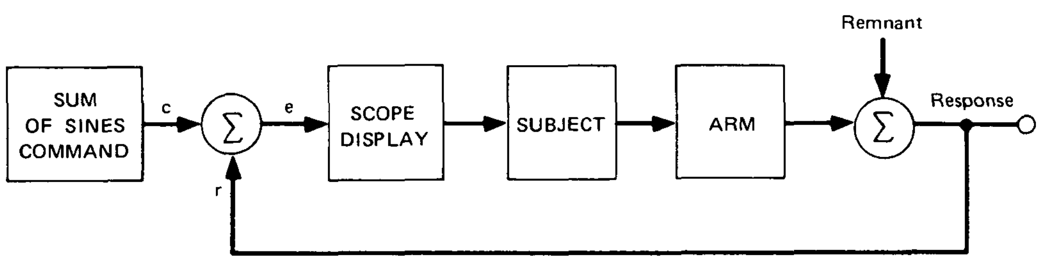


FIGURE 27 COMPENSATORY TRACKING SITUATION

out eight simultaneous tracking tasks (eight-dimensional tracking), only one dimension was used in this case. The one dimension of motion chosen was the axis of the forearm, as shown in Figure 28. This motion requires movement of two joints in the Rancho arm, and for this preliminary experiment, this motion is more representative of its ability than motions requiring only one joint to move. The error from the command position was displayed on an oscilloscope with its gain set so that ±0.6 inch corresponded to full deflection.

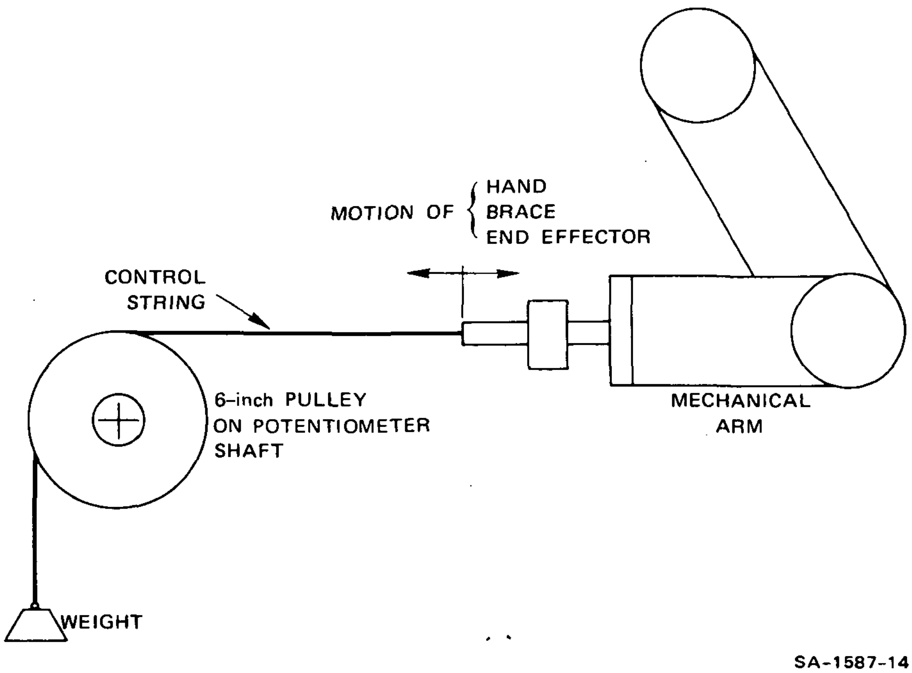


FIGURE 28 SIMPLE SYSTEM FOR MEASURING POSITION OF END EFFECTOR

The frequencies of the 16 sine waves in the command signal are given in Table 4. The bandwidth of the command signal as defined by McRuer et al. (1965) is approximately 1 Hz (or 6 radians per second). These high frequencies were chosen for the preliminary tasks so that accurate determination of the high frequency phase shift could be made, allowing the equivalent time delay, T_e, to be accurately determined.

In addition to analyzing the response at the 16 frequencies of the command signal to determine the operator's gain and phase shift, the response was analyzed at the 16 frequencies shown in Table 5 to determine the response remnant. It was felt that the operator's internally generated noise would be modified by both the jerkiness and inertia of the manipulator. The rms amplitude of this approximately Gaussian-distributed command signal was 0.84 inch.

Table 4

COMMAND SIGNAL SPECTRUM

Cycles per Run	Frequency (Hz)	Relative Amplitude
2	0.014	100
5	0.037	100
9	0.066	100
13	0.095	100
17	0.124	100
25	0.183	100
37	0.271	100
53	0.388	100
63	0.461	100
81	0.593	100
101	0.740	100
127	0.930	50
161	1.179	25
203	1.487	25
255	1.867	25
321	2.351	25

Table 5

REMNANT ANALYSIS FREQUENCIES

Cycles per Run	Frequency (Hz)
3	0.022
7	0.051
11	0.081
15	0.110
22	0.161
29	0.212
41	0.300
61	0.447
75	0.549
93	0.681
124	0.908
152	1.113
197	1.443
226	1.656
311	2.278
352	2.579

C. Subjects

Two subjects participated in the tests: JH was experienced using the Rancho arm, but SM had never used a manipulator before. There were, however, no obvious differences between subjects in their ability to track with the arm.

D. Procedure

Both subjects carried out each of the following tracking situations three times:

- Direct tracking--The subject tied a loop in the control string (see Figure 28) and put it around his index finger. The pickup potentiometer thus directly monitored the position of the subject's hand.
- Brace tracking--The control string was tied to the hand control portion of the Rancho brace. The pickup potentiometer thus measured the position of the brace as he moved it.
- Arm and brace tracking--The control string was tied to the end effector of the Rancho arm. The subject controlled the arm using the control brace.

Thus, there are three levels of control that would allow comparison of tracking performance with and without the brace to find limitations imposed by the brace, and with and without the Rancho arm to find limitations of the manipulator.

The tracking runs were 145 seconds long, including an 8.5-second warm-up time, during which the command signal was generated but no data were taken. The two subjects alternated, one running the other. The sequence Direct, Brace, Arm ... was repeated three times by each subject.

E. Results

The results of the tests are shown in the describing functions of Figures 29(a) and 29(b), which demonstrate that they are similar to previous tracking results with position control and can be fitted to the first degree with the simple crossover model of McRuer et al. (1965). Constants of the simple crossover model,

$$\frac{K e^{-sT_e}}{s},$$

fitted to the data are given in Table 6.

Table 6

CONSTANTS OF THE SIMPLE CROSSOVER MODEL

Vehicle	K (dB)	K	T_e (seconds)
Direct	2.2	1.29	0.156
Brace	-2.2	0.78	0.183
Brace and arm	-6.0	0.50	0.286

Using this numerical data and Figures 29(a) and 29(b), overall performance can be simply described in terms of changes in the open loop describing function. Such open loop changes for the separate components of the teleoperator system are given below.

1. Effect of the Rancho Brace

Comparing the (1) direct open-loop describing function and (2) brace open-loop describing function in Figures 29(a) and 29(b), it can be seen that the primary change brought by the inclusion of the Rancho

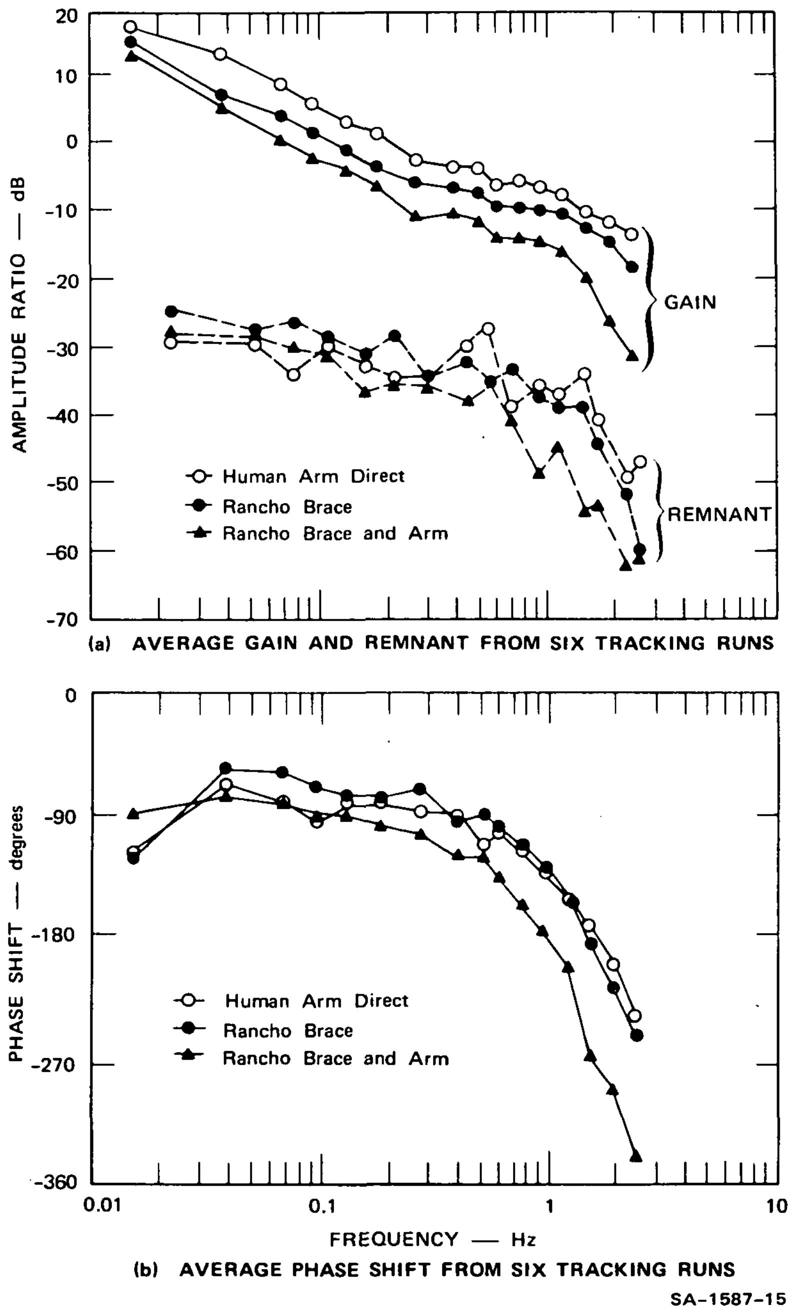


FIGURE 29 OPERATOR-VEHICLE DESCRIBING FUNCTION

control brace is a gain reduction of about 4.4 dB (or 40 percent), independent of frequency. The increase in equivalent time delay of 27 milliseconds (or 17 percent) has only a small effect on the high frequency phase shift.

2. Effect of the Rancho Arm

Comparing brace results with brace-and-arm results permits similar characteristics of the Rancho arm to be obtained. Primary effects are an additional large drop in gain of about 4 dB (or 36 percent) and a large increase in the high frequency phase shift accounted for by an increase in the equivalent time delay of 0.103 second (or 56 percent).

Scrutiny of the describing functions reveals a simpler description of the arm. If all three of the brace tracking curves (magnitude, phase, and remnant) are slid to the left (reduced in frequency by 40 percent), they fall on the corresponding brace-and-arm tracking curves. This corresponds to a simple time scaling and suggests that in compensatory tracking situations the addition of the Rancho arm simply slows the man-brace system down by 40 percent.

3. Effect of the Rancho Teleoperator

Comparing the direct and the brace-and-arm describing function allows the influence of the entire teleoperator system to be determined. Following the lines of the previous analyses, the teleoperator can be described as reducing gain by about 8 dB (or 62 percent) and increasing the equivalent time delay from 0.156 to 0.286 second (or 83 percent).

The remnant analysis of the response signal mirrors some of the changes in the describing function gain. At frequencies above 1 Hz, the curves are nearly identical, suggesting that the internally generated remnant is simply filtered by the mechanical characteristics of the

operator's own arm or the addition of the control brace and mechanical arm. Comparing remnant spectra from brace and brace-and-arm tracking shows the same uniform gain drop as noted in the corresponding describing-function gain curves, further supporting the mechanical filter hypothesis.

At low frequencies, however, there is greater remnant power in the runs with the Rancho brace and arm than with direct tracking. Below 0.5 Hz the additional equipment evidently adds to the operator's internal noise and, hence, contributes to increased tracking error.

F. Discussion

The characteristic gain and phase changes of the open loop describing function, together with the description of the operator's internally generated noise (remnant), can be used to compute his accuracy in actual closed loop operation. The changes in overall gain, equivalent time delay, and remnant power measured with different teleoperator systems can be used relatively to compare and rank them for performance.

Primary characteristics of the Rancho teleoperator system determined in this tracking study are:

- Uniform gain decrease, 8 dB
- Equivalent time delay increase, 0.13 second
- Low frequency human remnant increase, 9 dB.

Another useful characteristic is the equivalent time delay of the man-arm system equivalent to the parameter $\lambda = 1/T_e$ determined by critical tracking (Jex, McDonnell, and Phatak, 1966). From Table 6 we have $T_e = 0.286$ second and a fourth parameter: Critical pole $\lambda = 1/T_e = 3.49$ radians per second. The Rancho pole for critical stabilities is thus only half as large as the 6 to 7 radiants per second poles determined by Jex et al.

G. Further Work

This tracking approach to evaluating teleoperators is quite fruitful, and a refined tracking task should be developed. The command signal should be three dimensional to define a moving point in space, and its amplitude should be increased from the few inches of this preliminary study to cover a volume of a few cubic feet. In this way, all joints are called into play to move the end effector along the desired path, and difficulties of moving the arm (binding, shakiness, inertia, etc.) will be brought out in the results.

Simultaneous recordings of the position of the control brace end point and the end effector will allow the brace and arm describing function and remnant power to be determined simultaneously.

Several characteristics of teleoperator systems can be determined from these analyses. Variation in describing functions with different arms, braces, time delays, and viewing systems may also allow simple characterization of these complicated, often nonlinear systems.

Appendix A

INFRARED RANGE SENSOR

Appendix A

INFRARED RANGE SENSOR

For both automatic and manual control, it is important to be able to sense objects in the neighborhood of the end effector before a collision with them causes damage to the arm or the object struck. The critical area of interest in normal manipulation tasks is the field immediately in front of the end effector ranging from 1/2 inch to about 5 inches away from the tip of the tongs. A range sensor based on infrared triangulation that provides this information is shown in Figure A-1.

The basis of the sensor is overlapping fans of light emitters and light sensors. Four light-emitting diodes provide four diverging light beams that illuminate objects lying in front of the end effector. Four phototransistors, in symmetrically opposite positions, sense incoming light from four converging beams. An object located at any of the intersection points of the two fans causes light to be diffused or reflected from one of the light-emitting rays to one of the light-collecting rays.

The intersecting light beams are separated and identified by amplitude modulation at different frequencies, as indicated in the block diagram of Figure A-2. The separate light-emitting diodes are modulated at 2.0, 2.4, 2.8, and 3.2 kHz, respectively. Similarly, signals from the separate phototransistors are individually filtered by separate synchronous detectors at the same frequencies. In this way, each of the possible intersecting light beams can be individually identified.

Initial development consisted of a bench-top design with a single light-emitting diode and a single phototransistor. It was found that, with high power (1 watt) light-emitting diodes, an external lens for

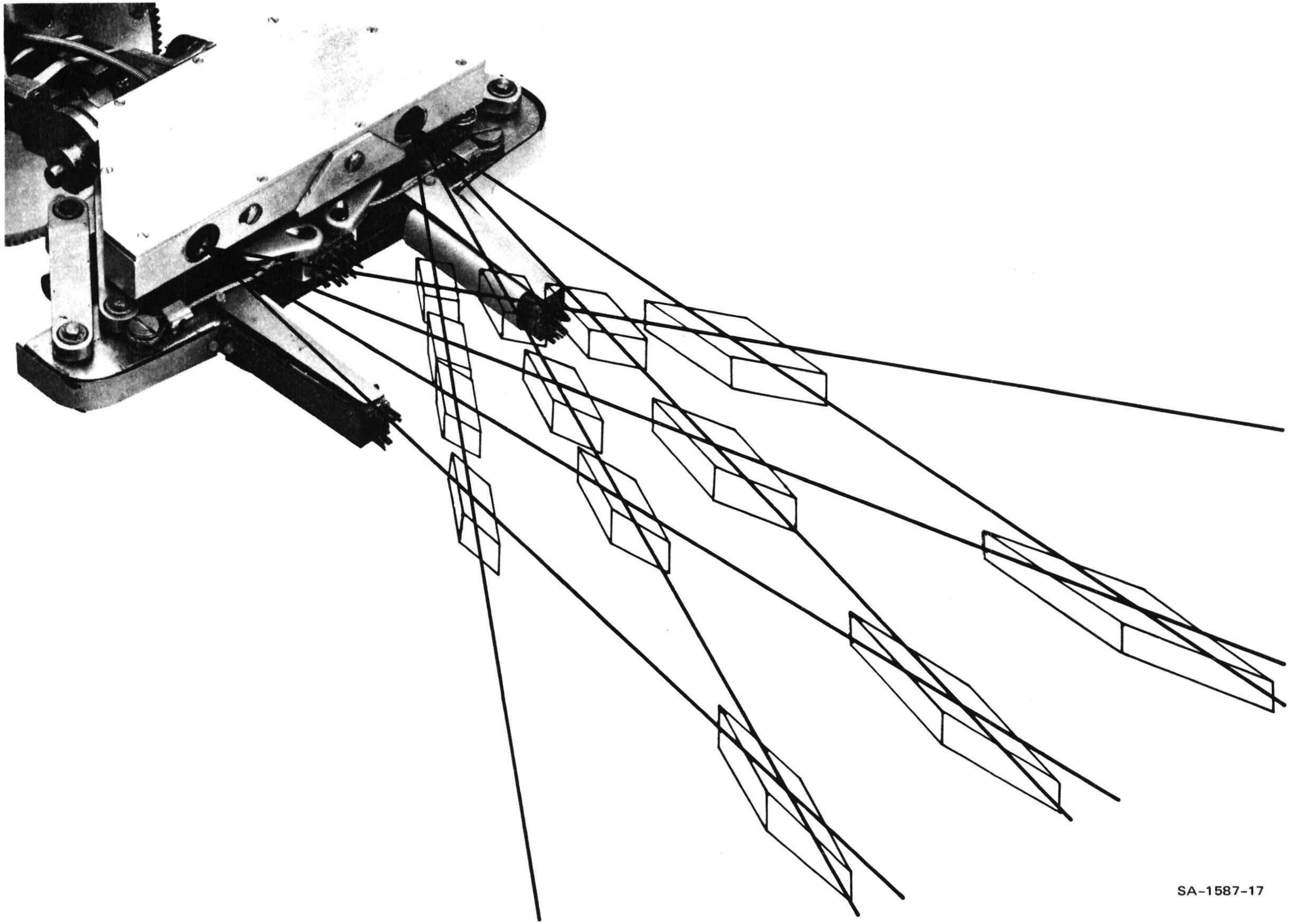


FIGURE A-1 FIELD OF VIEW OF RANGE FINDER, IN PERSPECTIVE

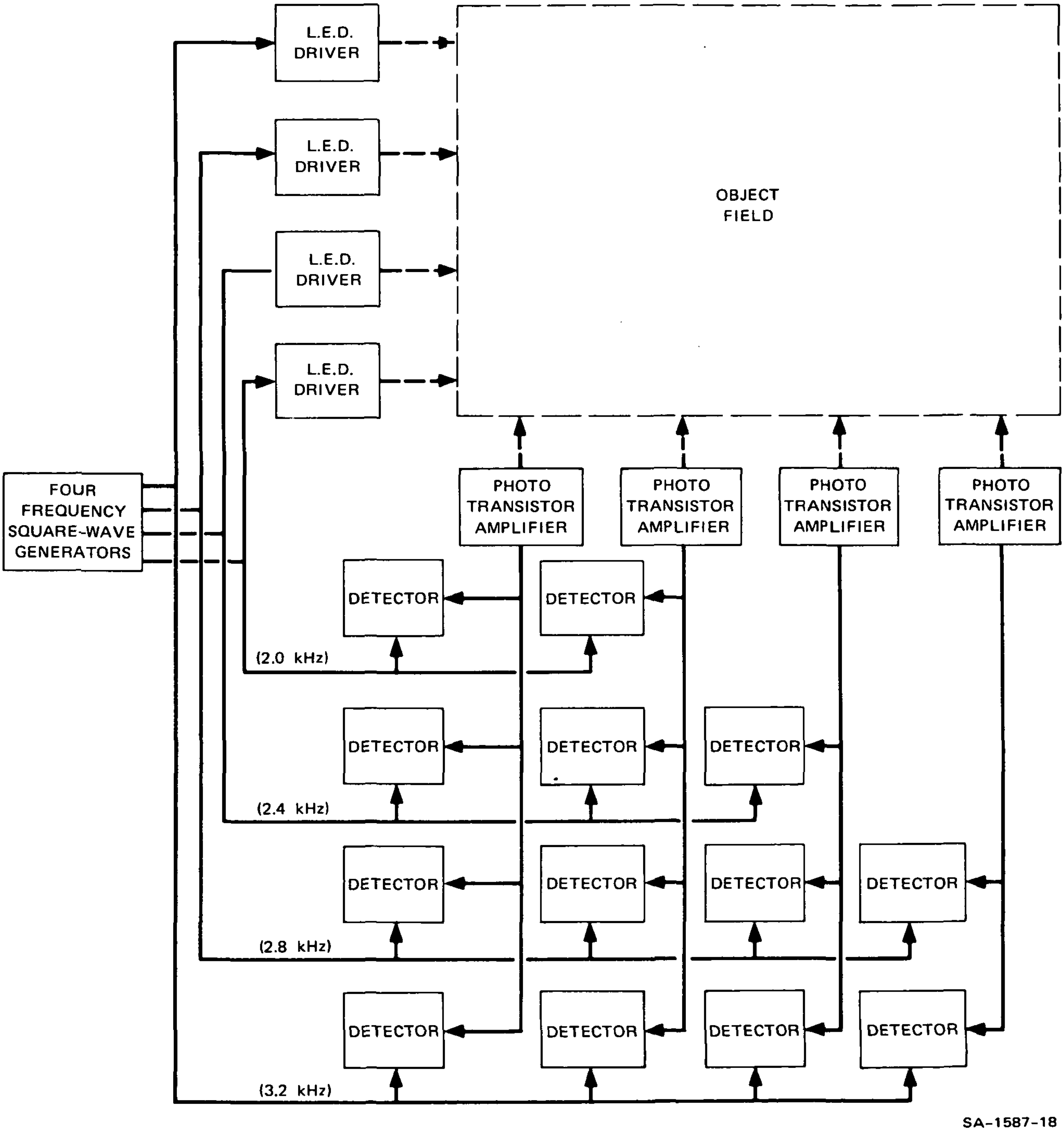


FIGURE A-2 BLOCK DIAGRAM OF OPTICAL RANGE SENSOR
(DET is synchronous detector plus threshold detector.)

collimating light beams, an infrared filter (Wratten 87C) in the light-collecting path, and care in grounding and preamplification, even very black or reflecting objects could be reliably detected. Only polished metal parts and mirrors escaped detection.

The current status of the range sensor is represented by Figures A-1 and A-3. The device attaches to the top of the Rancho end effector and is mechanically complete. Single electronic components consisting of a light-emitting diode (L.E.D.) driver, phototransistor driver, synchronous detector, and threshold detector have been designed and debugged on the single-beam bench-top model. An electronic package of these components for the 4 X 4 range sensor has not been completed, and it is estimated that about one man-month of additional work would be involved to turn the sensor into an operating system.

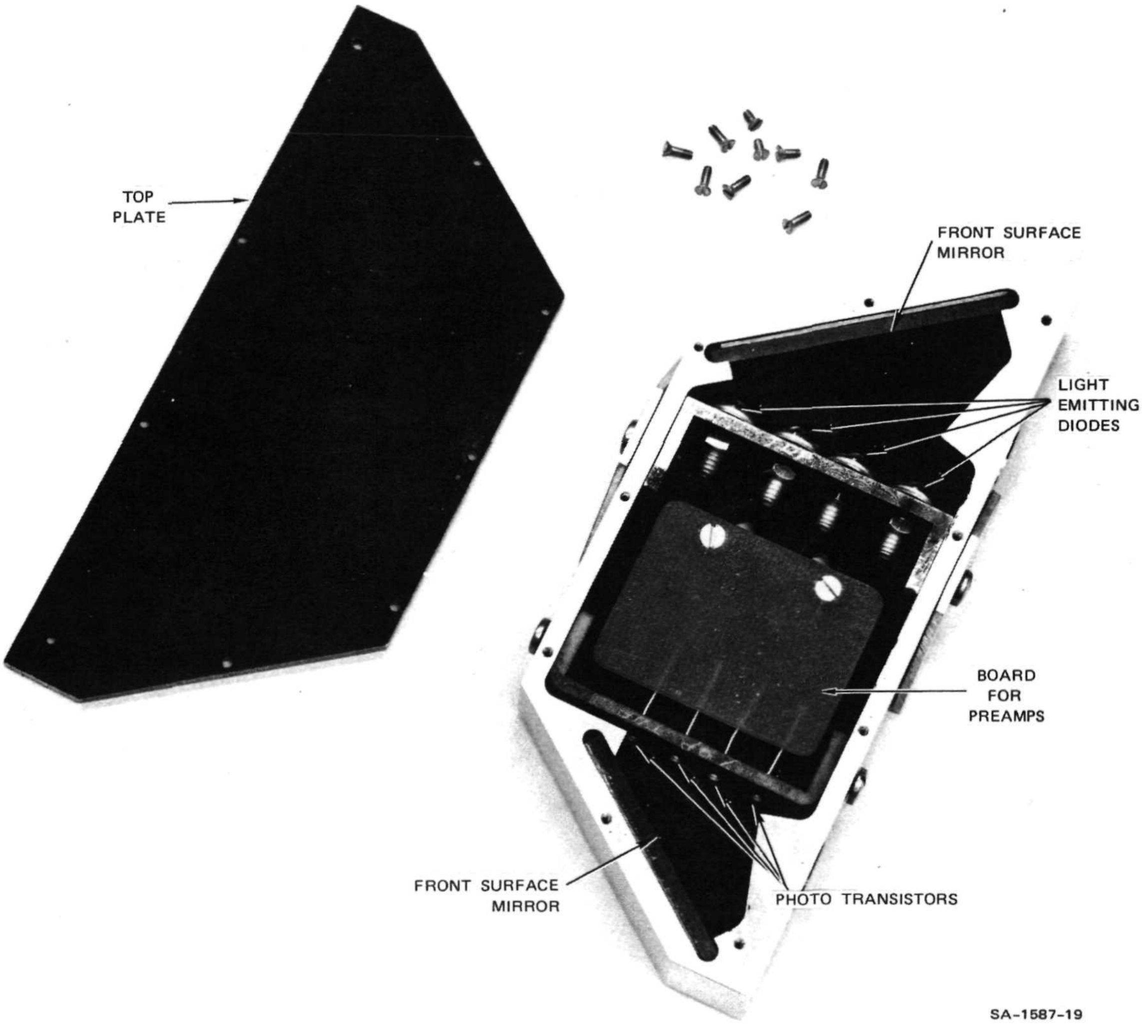


FIGURE A-3 INTERNAL CONSTRUCTION OF THE RANGE SENSOR

Appendix B

COMPUTER INTERFACE FOR TACTILE INFORMATION

Appendix B

COMPUTER INTERFACE FOR TACTILE INFORMATION

To carry out automatic operations based on touch, the control computer must be able to query the touch-sensing systems built into the end effector previously described in Section V. To make this possible, considerable effort during the project was spent designing and constructing an interface to the LINC-8 arm control computer. The interface permits the two 6 × 24 binary touch patterns from the tong pads and the 21 external binary sensors to be input to the computer through a single program call. The interface is basically a parallel-to-serial converter that inputs the tactile picture shown in Figure B-1 to a designated area of the computer's memory. Detailed mappings of the external touch sensors are shown in Figures B-2 and B-3.

Individual sensors are tested using the series of masks described in Section III and illustrated in Figure B-1. There is a row mask and a double word column mask for testing of the sensing pads. A double word mask is used to specify any of the external sensors.

The implementation of the parallel-to-serial converter in hardware is shown in Figure B-4. The 12 × 32 bit matrix is composed of 12 32-bit shift registers in parallel. Following the JAM signal from the computer, the contents of the 309 parallel lines to the touch sensors are entered in the shift registers. The next 32 consecutive READ and CLOCK signals generated by the computer shift and transfer the contents of the shift registers, 12 bits at a time, into the LINC-8 accumulator and subsequently to an area of memory.

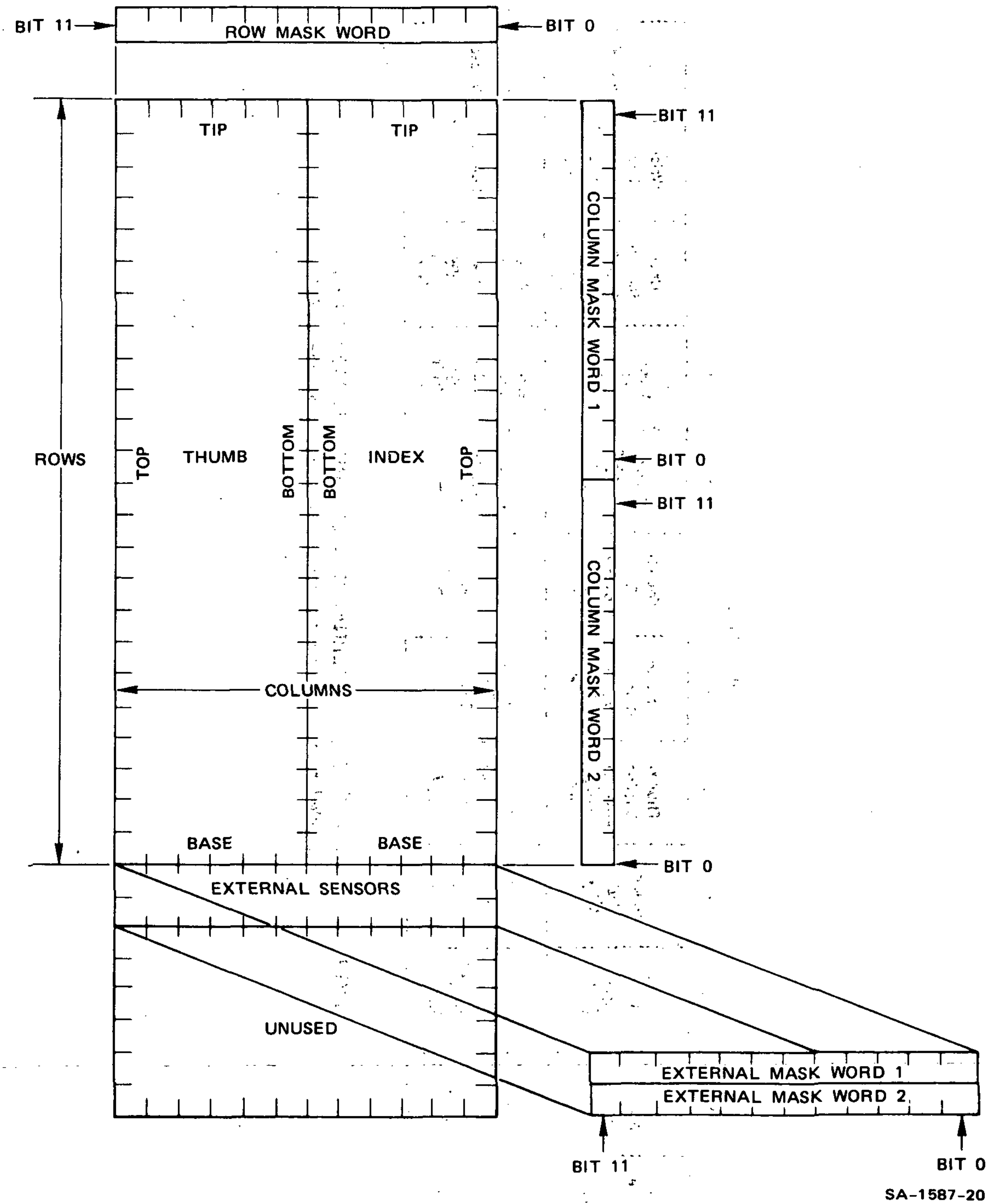


FIGURE B-1 MAPPING OF TACTILE SENSORS INTO 12 BY 32 BIT SENSOR MATRIX. Sensor test mask words are shown in relation to the areas tested on the jaws. The mask words are not part of the sensor matrix.

BIT NUMBER

11	10	9	8	7	6	5	4	3		1	0
Index Outside Distal 15	Index Bottom Distal 16	Index Top Distal 17	Index Tip 18	Thumb Outside Proximal 12	Thumb Bottom Proximal 13	Thumb Top Proximal 14	Index Knuckle 19	Webb of Jaw 9	Wrist Bottom Thumb Side 10	Wrist Top Thumb Side 21	Unused
Index Outside Proximal 5	Index Bottom Proximal 6	Index Top Proximal 7	Thumb Outside Distal 1	Thumb Bottom Distal 2	Thumb Top Distal 3	Thumb Tip 4	Thumb Knuckle 8	Wrist Bottom Index Side 20	Wrist Top Index Side 11	Unused	Unused

SA-1587-21

FIGURE B-2 MAPPING OF EXTERNAL SENSORS INTO SENSOR MATRIX. Circled numbers in boxes correspond to those of Figure B-3.

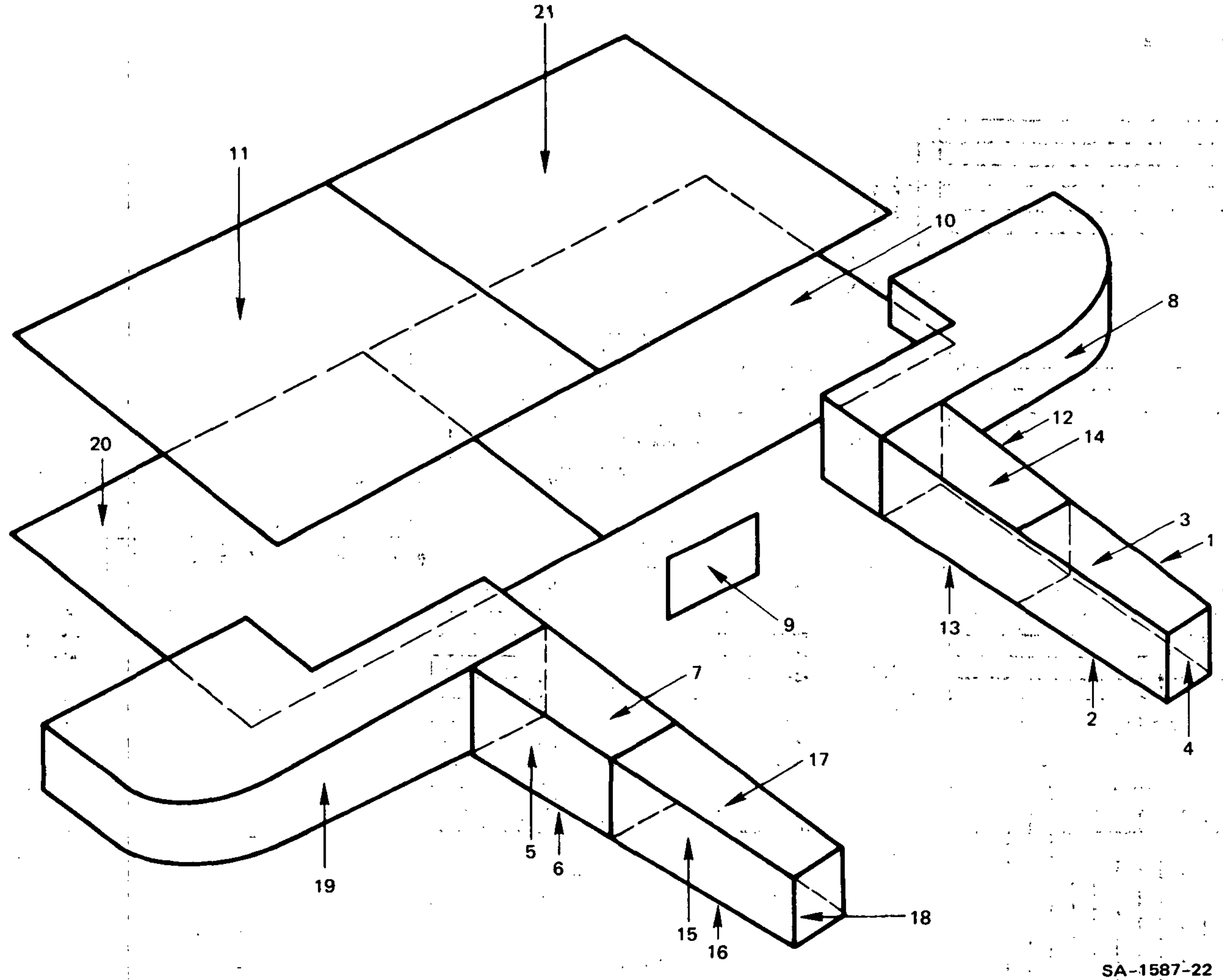


FIGURE B-3 LOCATION OF EXTERNAL TOUCH SENSORS ON END EFFECTOR. Numbers correspond to those on Figure B-2.

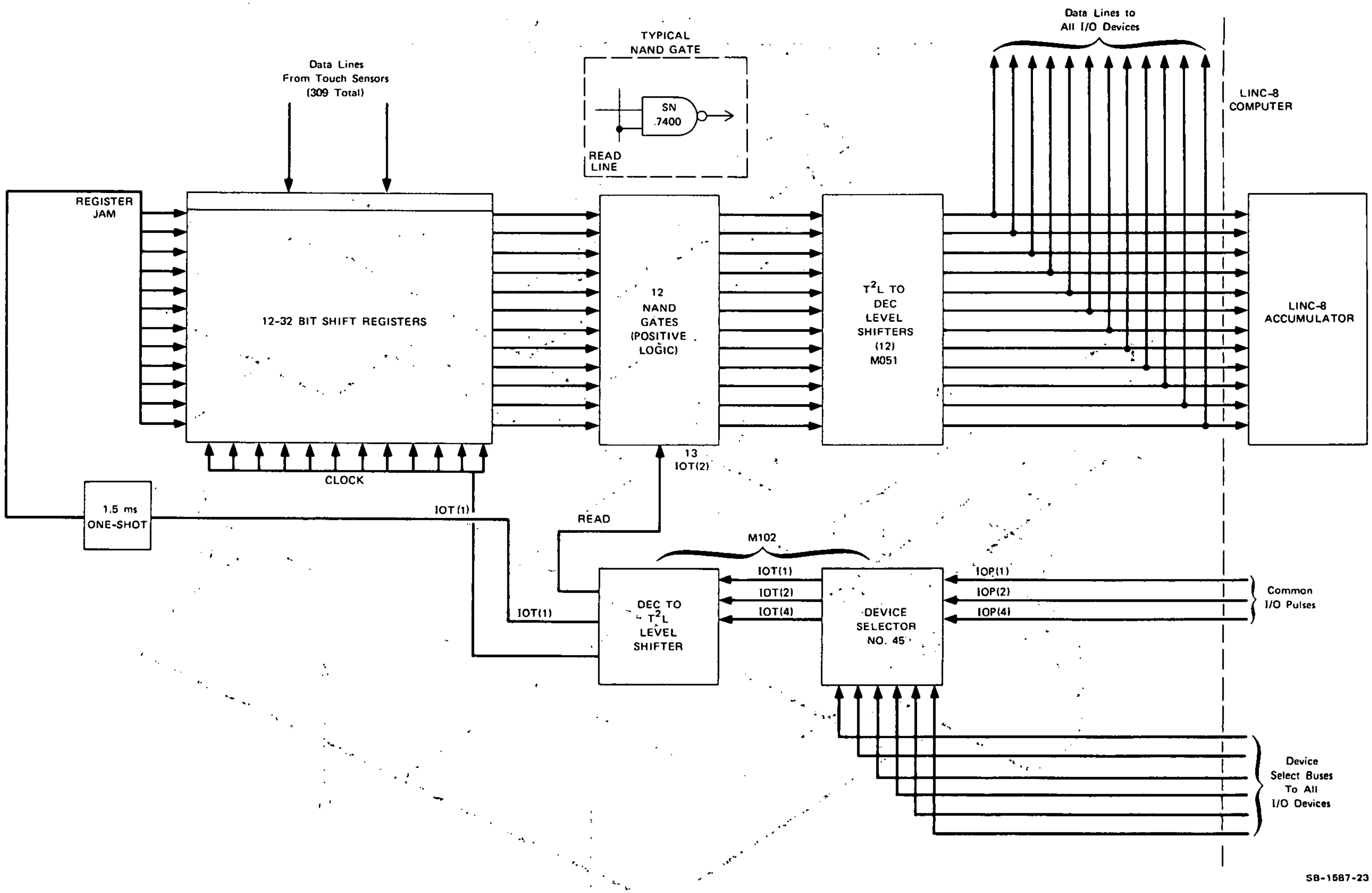
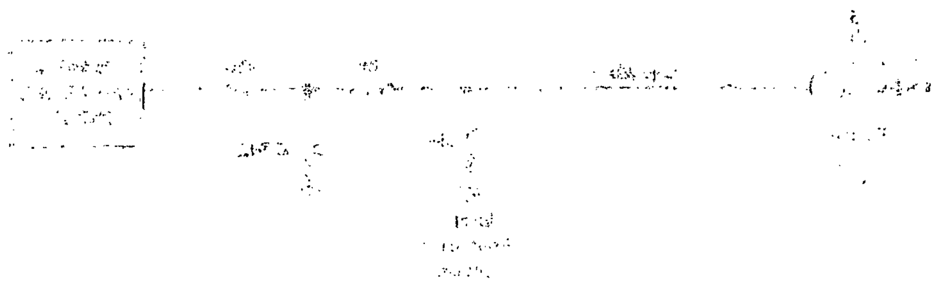


FIGURE B-4 TACTILE INTERFACE SYSTEM DIAGRAM



For ease in programming, the series of instructions that cause the touch information to be read in is allocated to the resident. A single instruction of the form LUCI; AHO 3 to the resident (named LUCIFER) causes the necessary control signals to be generated and the internal bookkeeping to be carried out so that the contents of the touch matrix are moved to the internal location desired.

Each of the 32 input lines is protected by the circuit shown in Figure B-5; the detailed design of the 32-bit shift registers are shown in Figure B-6.

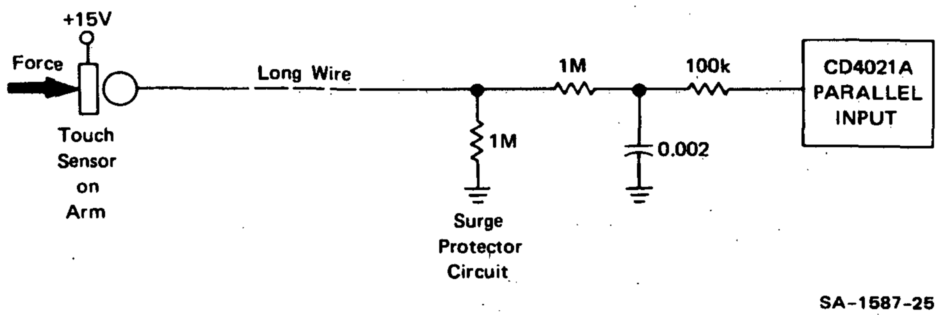


FIGURE B-5 INDIVIDUAL TOUCH SENSOR TO SHIFT REGISTER CONNECTION

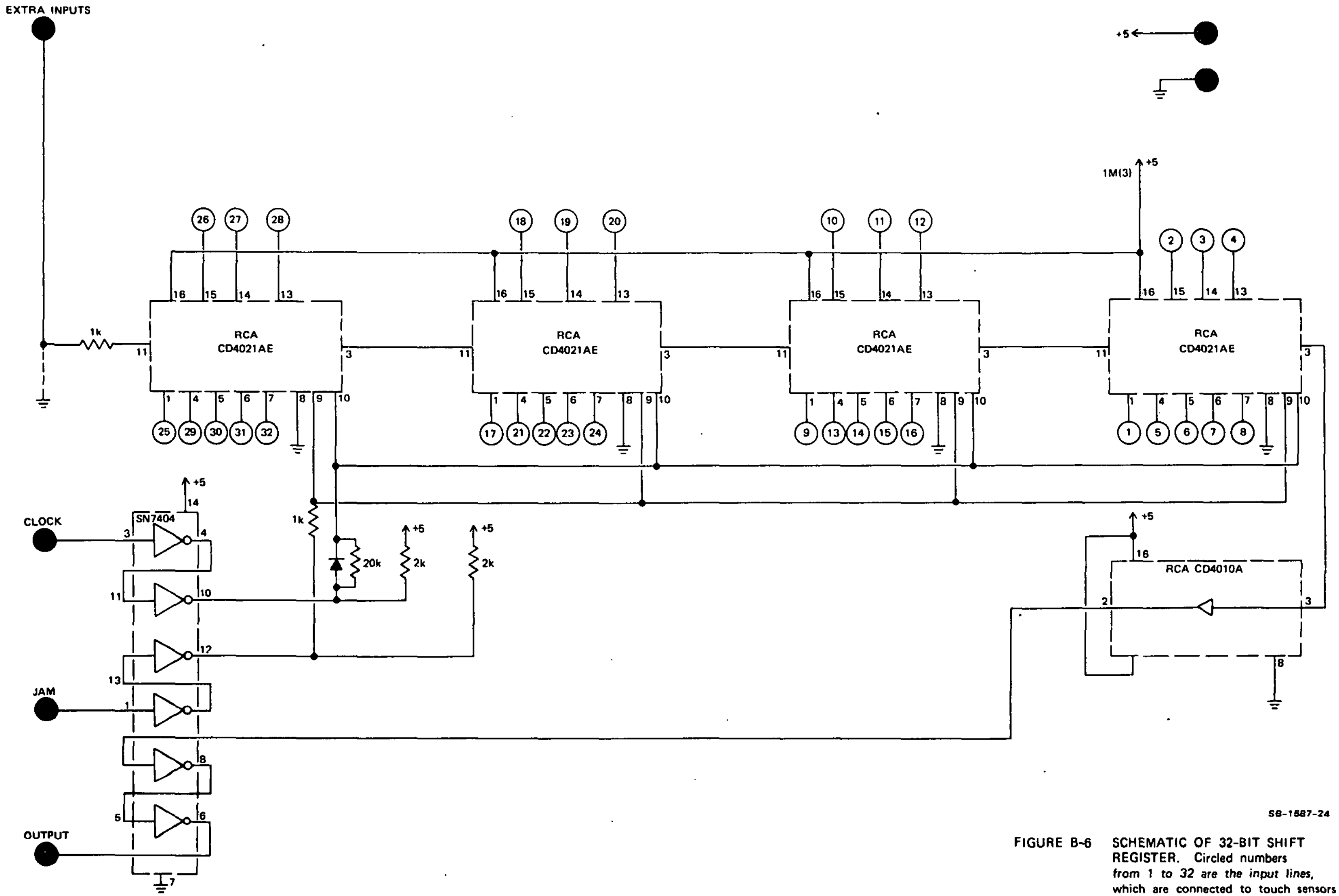


FIGURE B-6 SCHEMATIC OF 32-BIT SHIFT REGISTER. Circled numbers from 1 to 32 are the input lines, which are connected to touch sensors through isolating circuits.

Appendix C

PROPORTIONAL CONTROL SYSTEM

Appendix C

PROPORTIONAL CONTROL SYSTEM

Modifications of the proportional control system were made that increased the positioning accuracy of the Rancho arm. With the motor driver circuitry of Figure C-1, the end effector has worst-case absolute accuracy of about 1 inch (arm outstretched) and nominal accuracy of 1/2 inch. Repeatability, however, as in reacquiring a previously obtained position is better than 0.2 inch. This measurement is made by recording the joint angles of the arm using the DEFINE command and repeatedly returning using the MOVE TO command. An integral part of the control system is the ramp generator of Figure C-2, which supplies the high frequency ramp signal for converting the proportional error signal to a variable width pulse.

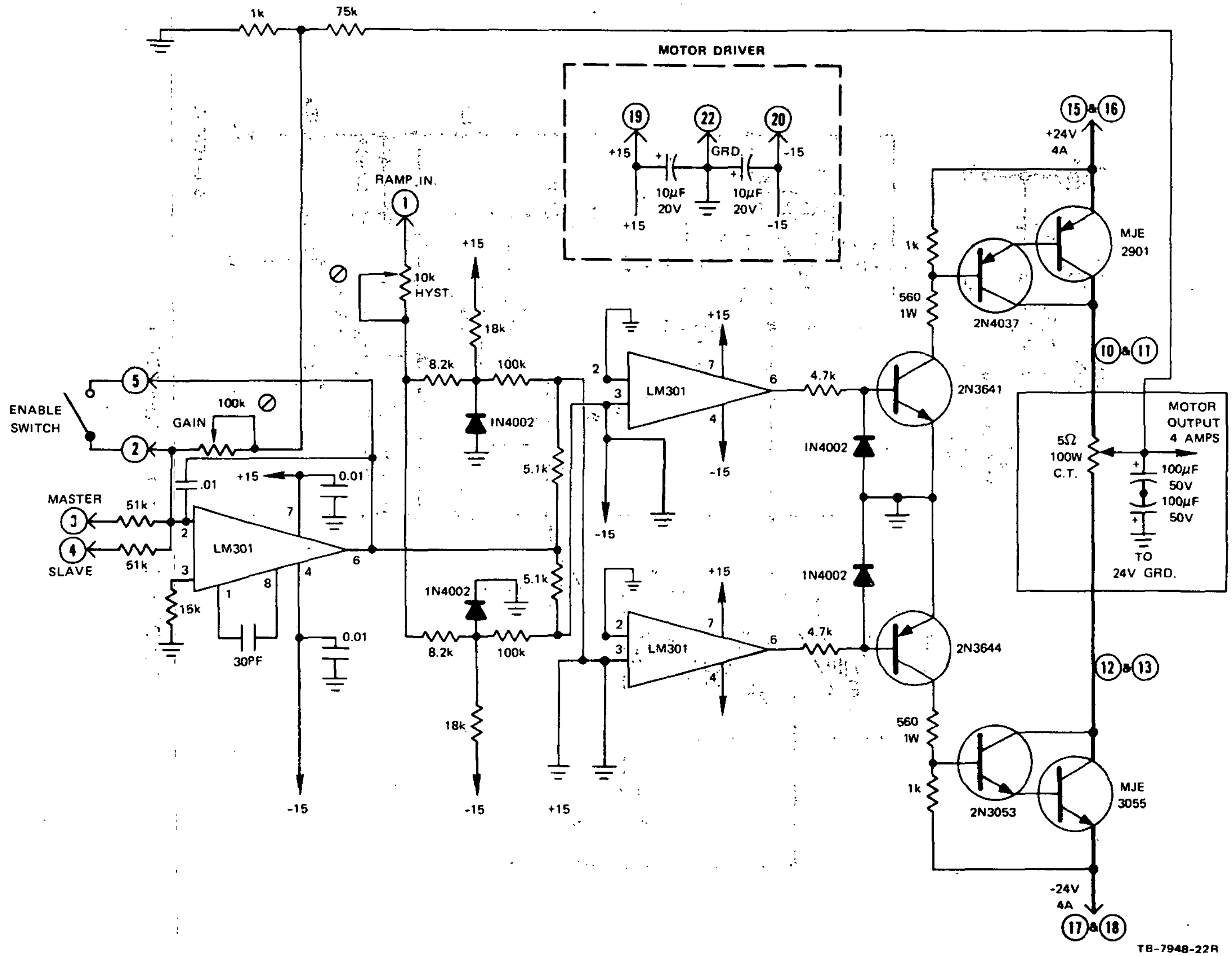


FIGURE C-1 MOTOR-DRIVER CIRCUIT IN PROPORTIONAL CONTROL SYSTEM

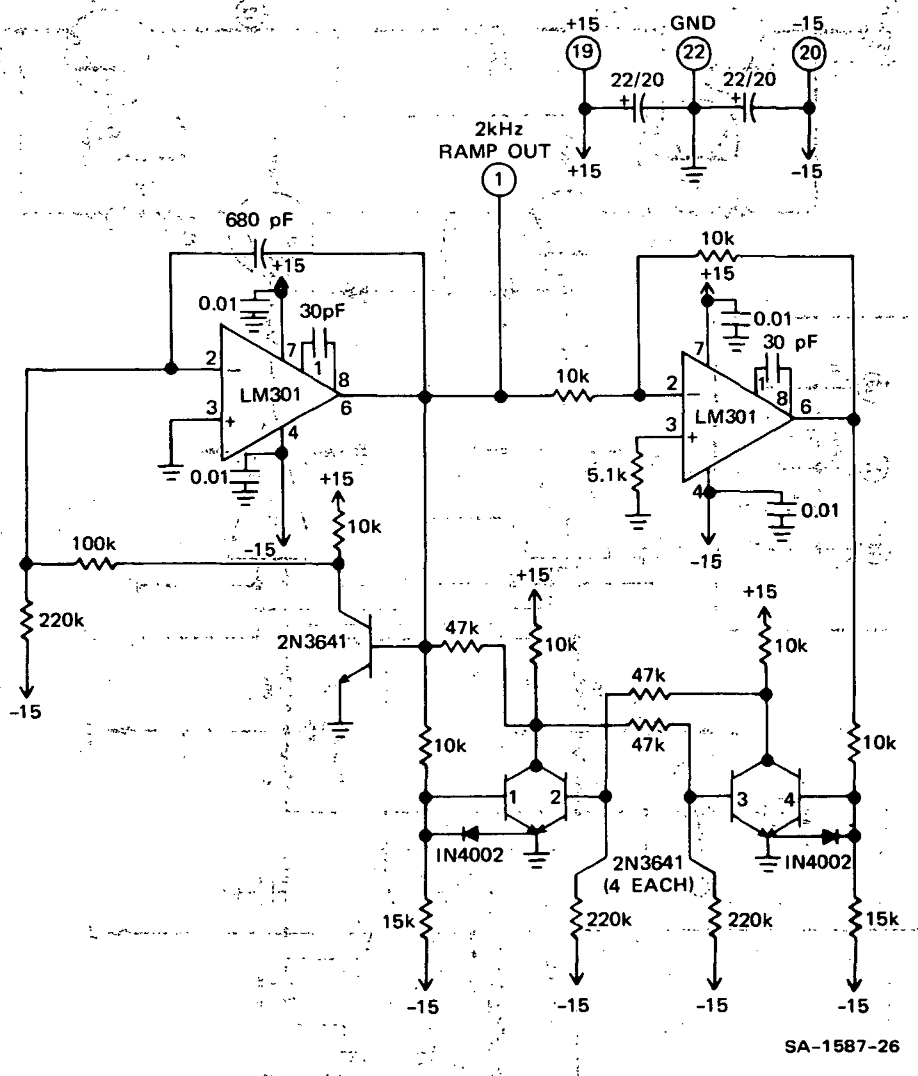


FIGURE C-2 RAMP GENERATOR

Appendix D

CHART RECORDER SETUP FOR MONITORING HUMAN MANIPULATION PERFORMANCE

A multiple-channel chart recorder has been used to study manipulation tasks under different conditions of time delay, automatic operations, sensory feedback, and viewing conditions. The four channels of the recorder monitor instantaneous master position, slave position, power consumed, and task status, as indicated in Figure D-1.

Two networks of summing resistors used to monitor master and slave positions are perhaps more useful than the 14 channels required to monitor the joints individually. The number and duration of moves in a particular task can be counted directly from the graphical output. In addition, the computer performance monitor (described in Section IV) follows all 14 joints for possible moves, and it signals a move on the corresponding event marker (master or slave). In this way, even when the highly improbable move with self-canceling joint sums is made, the performance monitor signals the move on the event marker.

For measuring the power consumed during a test run, a practical indicator of performance, the current sensor of Figure D-2 was implemented. Instantaneous power is determined by multiplying the current to the servo system by the 24-volt servo EMF. To compute the energy consumed during the task, the performance monitor reads the current sensor every 1/30th second and accumulates it as a total. In this way, the total number of ampere-seconds (converted to watt-seconds by multiplying by 24) is obtained and printed out at the termination of the test run. A typical recording for the task of picking up an object from a flat table is shown in Figure D-3.

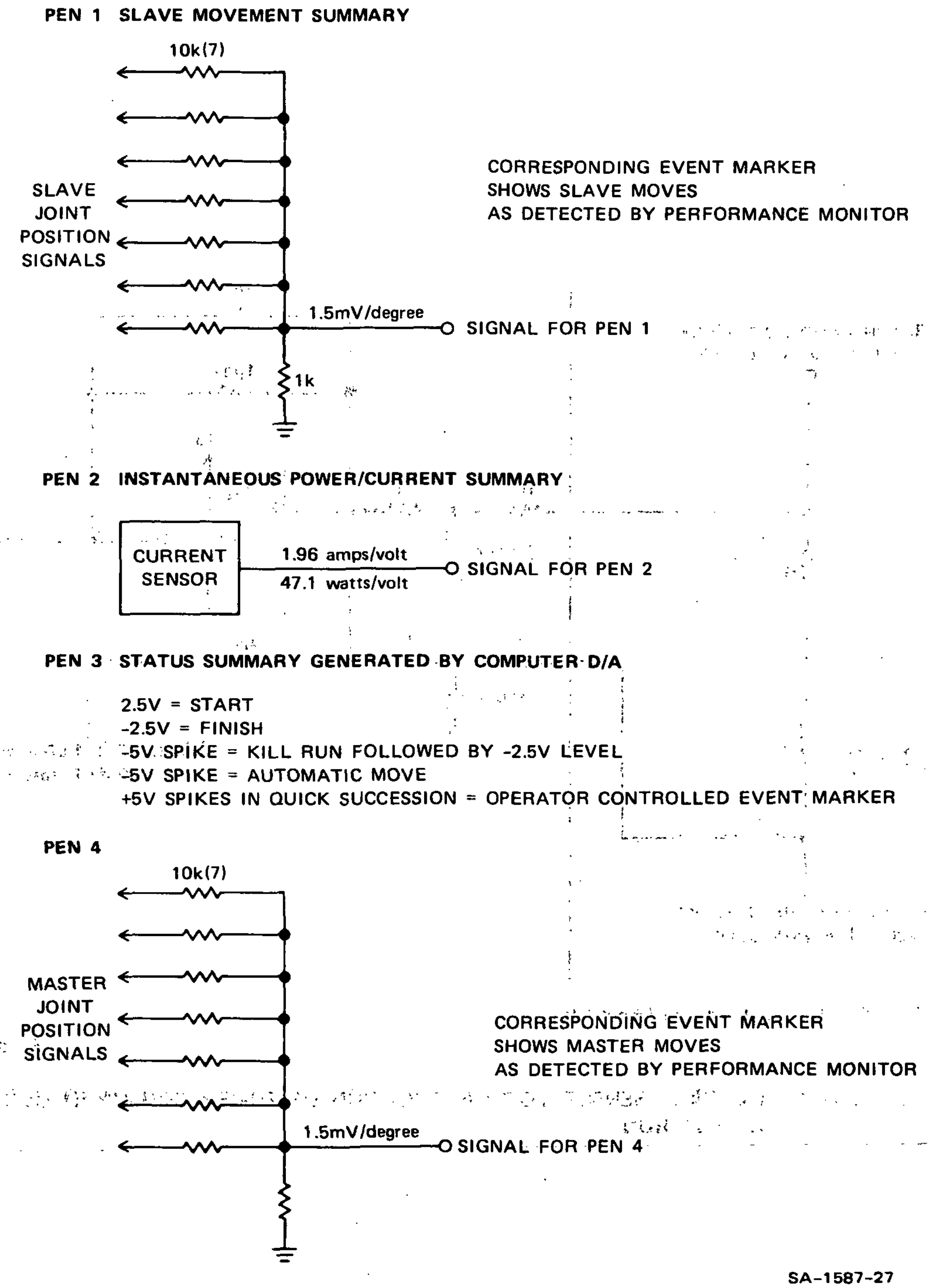
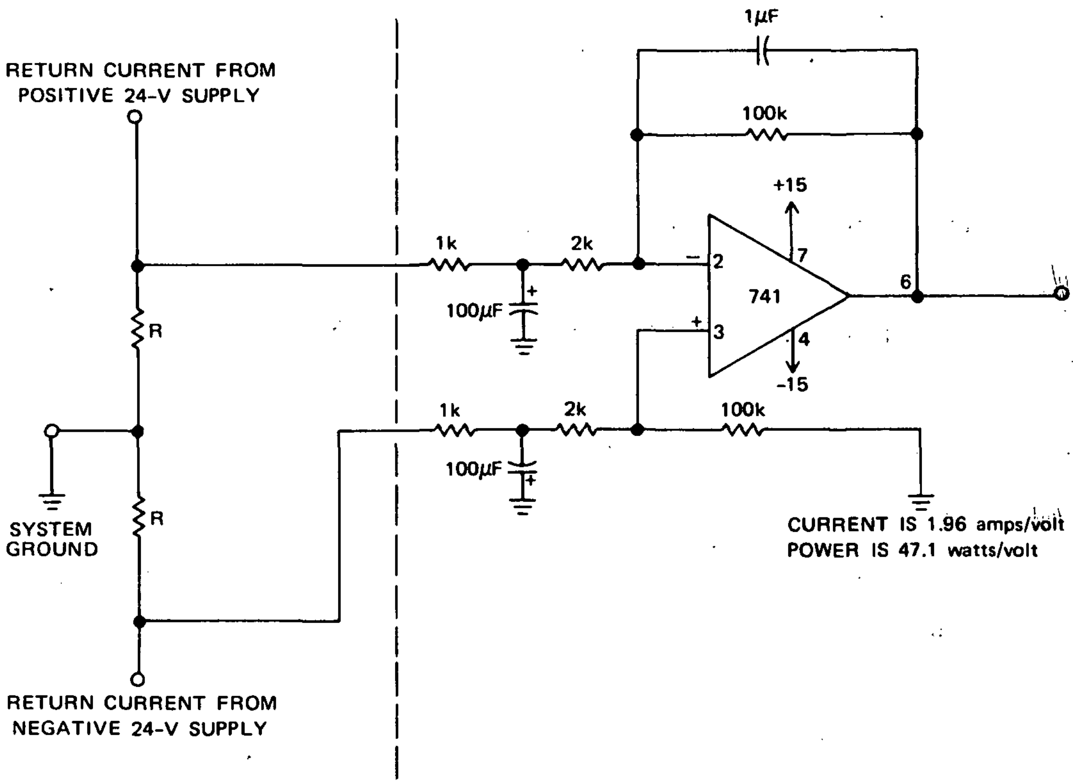


FIGURE D-1 CONNECTIONS TO CHART RECORDER FOR STUDYING MANIPULATION TASKS



R is 3.5 feet of 16 gauge wire (about 0.017Ω).

SA-1587-28

FIGURE D-2 CURRENT SENSOR FOR CALCULATION OF POWER CONSUMED DURING A TEST RUN

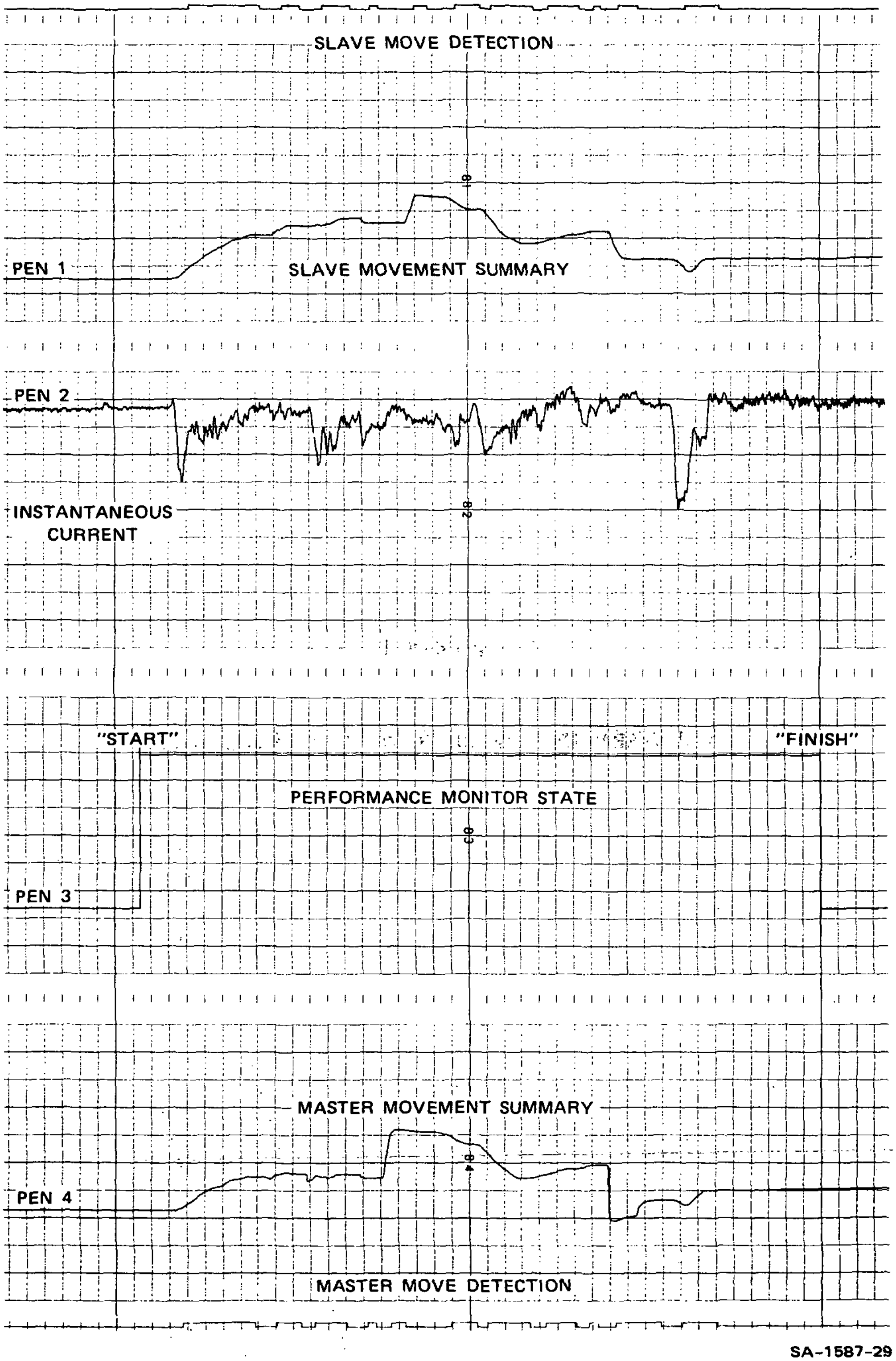


FIGURE D-3 CHART RECORDER FORMAT FOR MONITORING PERFORMANCE

Appendix E

ARM OPERATING SYSTEM OVERALL ORGANIZATION

Appendix E

ARM OPERATING SYSTEM OVERALL ORGANIZATION

As has been previously described, the servo loop is completed within the computer. This is advantageous in that the gains may be readily changed to suit the task at hand, or the movement of the arm may be interrupted at any time in response to sensor readings or operator interrupts. In order to accomplish this, the arm system must function in real time, and the servo system function must be accomplished at least 30 times per second. Additionally, the computer must be capable of performing sensor evaluations, providing teletype functions to the operator, servicing the remote station, and monitoring the performance of the arm. Thus, the resident arm computer system has been organized to run cyclically at 30 Hz, performing each of the necessary functions once per cycle. Any single function cannot require more than the allotted amount of time, or the system fails. At present, the idle time (that time during which the computer may perform additional tasks) varies between 12 and 26 milliseconds per cycle. As functions that are more time consuming are added to the system capabilities, we will have to break them into smaller parts so that real time may be maintained; this has not as yet become necessary.

In order to aid in timing and debugging, the various major functions are delineated by various voltages output to an oscilloscope. This proves also to be of value in following general program logic. The cyclic program structure described above is illustrated in Figure E-1. The corresponding oscilloscope presentation is shown in Figure E-2.

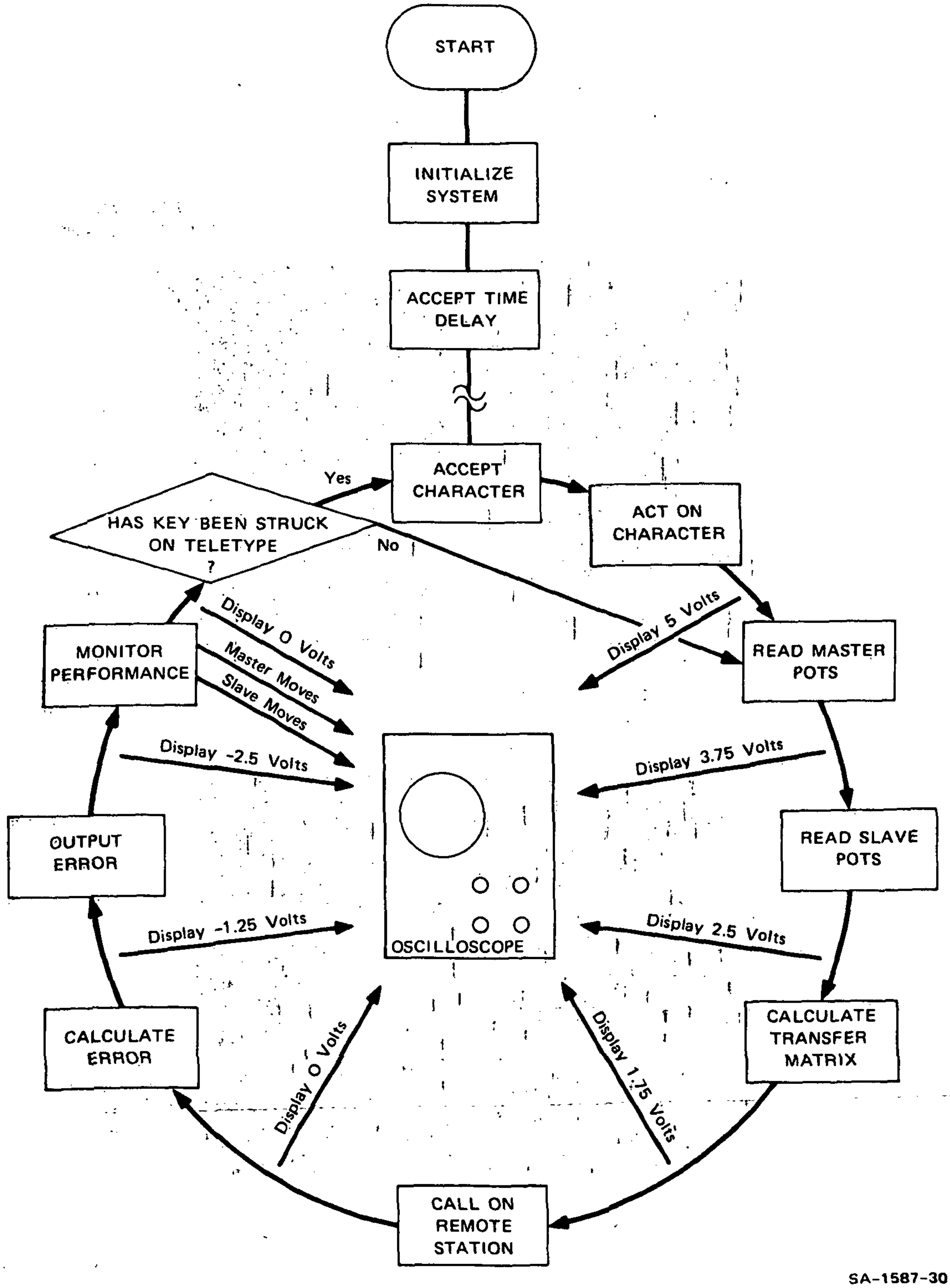


FIGURE E-1 OVERALL SYSTEM FLOW CHART

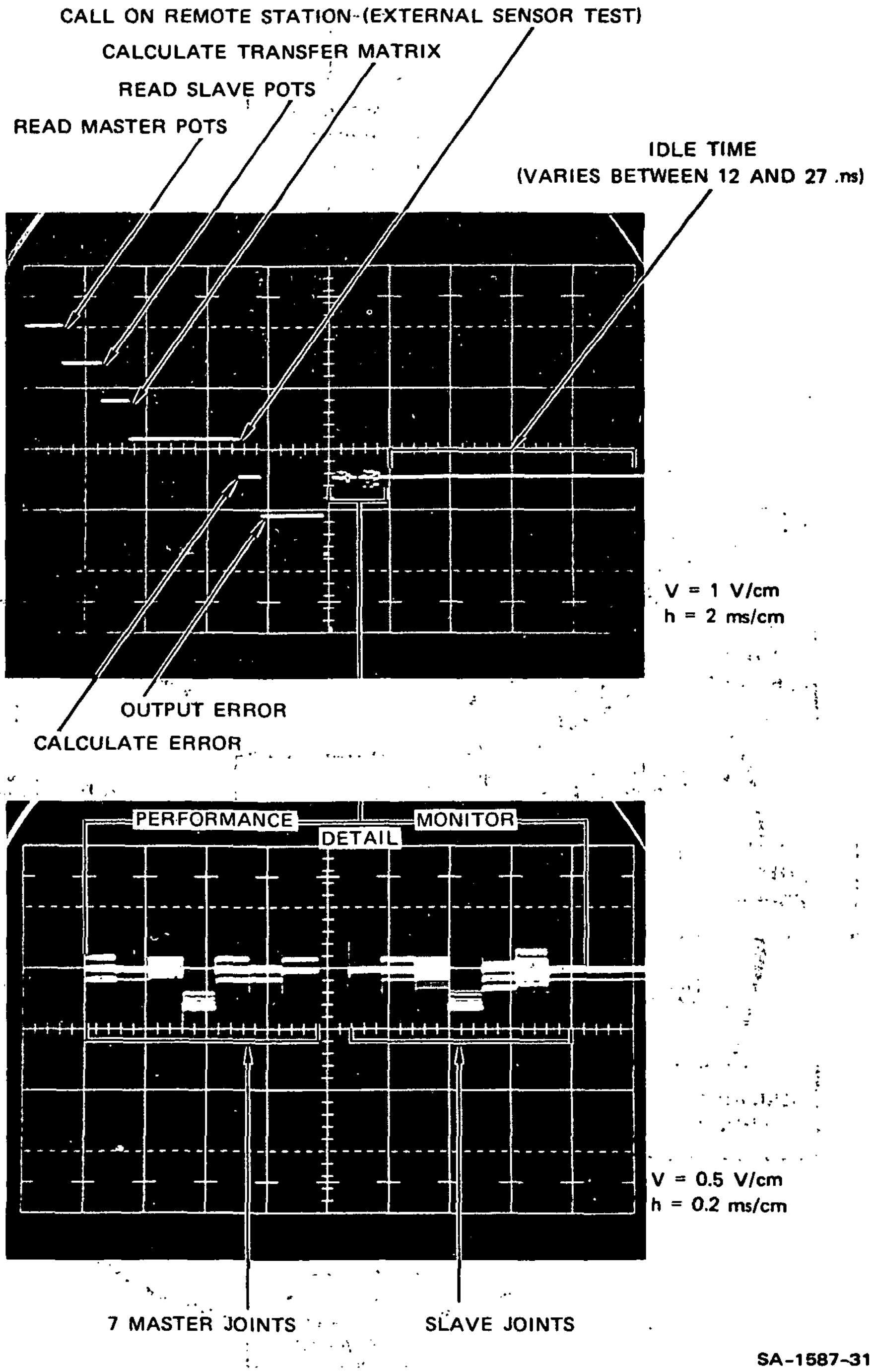


FIGURE E-2 OSCILLOSCOPE DISPLAY OF THE MAINTENANCE OF REAL TIME (Joint 4 is moving.)

Appendix F

DESIGN OF TOUCH FEEDBACK EVALUATION EXPERIMENTS

Appendix F

DESIGN OF TOUCH FEEDBACK EVALUATION EXPERIMENTS

To evaluate the usefulness of the touch sensing and feedback systems developed in this study, a series of experiments were designed. These experiments answer questions about the value of touch feedback with:

- Different viewing conditions
- Various amounts of time delay
- Tasks of varying difficulty.

In all three of these cases, intuition would lead us to expect that touch feedback would help the operator carry out certain parts of his task.

The important questions to be answered are those that can be answered objectively rather than subjectively. For example, we want to know what are the savings in task time, or what is the reduction of drops and fumble in a given situation. In conjunction with the on-line performance measuring system described in Section IV, the following series of three experiments has been designed. Each experiment considers one of the above three variables. The interaction of one of these variables with three types of tactile display conditions is the subject of an experiment. For all three of the experiments, the tactile display conditions are:

- F_o--No feedback. No information from the touch sensors is presented to the operator.
- F_t--Tactile feedback. The tactile display system consisting of two bimorph displays and air-jet contact display will be provided to the operator.
- F_v--Visual feedback. The CRT moving jaw display of the touch sensors will be provided to the operator.

The operator's task in each case is to pick up an object (a block or latch) and move it away. Performance measurements are made using the capabilities of the on-line computer program described in Section IV. Details of each experiment are given below.

1. Experiment I: Variable Viewing Conditions

The three viewing conditions of this experiment are defined below.

- V_d--Direct viewing. The operator views the scene directly from a position about two meters away.
- V_{tv}--TV viewing. A closed-circuit, broadcast-quality TV system is imposed.
- V_{tv+n}--Noisy TV viewing. Same as T_{tv}, except that a white noise is added to the video (S/N = 0 dB).

a. Subjects

Two male subjects were paid for their services. Both subjects practiced all conditions of this task until their task completion times stabilized. Each subject participated in this experiment approximately two hours per day.

b. Procedure

The three viewing conditions and three feedback conditions define a 3 X 3 factorial experimental design, as shown in Figure F-1. Each cell of the design, representing a single viewing and feedback condition, consists of 10 repeated block pickups. To ensure that the order in which these nine conditions are carried out does not bias the experimental results because of continuously improving performance, the conditions are ordered using a Graeco-Latin square technique. In this way, practice effects will not bias any viewing or feedback condition.

FEEDBACK CONDITIONS

		F_o	F_t	F_v
VIEWING CONDITION	V_d	1	6	8
	V_{tv}	5	7	3
	V_{tv+n}	9	2	4

SA-1587-32

FIGURE F-1 DESIGN FOR TOUCH FEEDBACK EVALUATION EXPERIMENTS

When the viewing and feedback conditions are set up, the experimenter starts the on-line performance monitor. When the computer has initialized the appropriate file and is ready, a bell is rung signaling the subject to begin. When he has successfully retrieved the block and brought it back past a marker post, the experimenter signals the computer to stop monitoring and to print out run time and power consumed.

c. Analysis

The performance indices are entered into a 3 × 3 analysis of variance (factorial design) to determine the effects of the main variables and their interaction. The analysis also yields the residual variance for testing hypothesis about the individual conditions.

2. Experiment II: Variable Time Delay

This experiment is identical in design to Experiment I, except that instead of three viewing conditions, three time-delay conditions are substituted. These time-delay conditions are defined below.

- T_o--No time delay
- T_1--One-second time delay
- T_3--Three-second time delay.

The operator's task in Experiment II is to pick up a block with direct viewing under all nine combinations of feedback and time delay. The experimental design and analysis are identical to those of Experiment I.

3. Experiment III: Obscured Viewing

This experiment is identical in design to Experiment I, except that the three different obscured viewing conditions described below replace the three viewing conditions.

- O_o--No obscuration. Retrieve a latch from a hole with direct viewing.
- O_p--Partial obscuration. Same as O_o except that approximately half of the latch is out of view behind a metal plate.
- O_t--Total obscuration. Same as O_o except that latch is entirely out of view behind the metal plate.

The experimental design and analysis are identical to those of Experiment I.

Appendix G

DESIGN OF SUPERVISORY CONTROL EXPERIMENT

Appendix G

DESIGN OF SUPERVISORY CONTROL EXPERIMENT

1. General

This experiment is designed to determine the advantages of automatic control features based on touch sensing in a computer-augmented teleoperator system. The task is picking up several objects in the work space and depositing them in a common receptacle.

The supervisory control system is described in Section II and summarized in Figure G-1 under different conditions of transmission delay. The use of the GROPE and DROP programs in this task is described in Section III, as is the method of calling these programs into play.

2. Experimental Design

The experiment is arranged in a $4 \times 5 \times 2$ factorical design, as shown in Figure G-1. Each cell in the design represents a performance characteristic measured on two practiced subjects in fine replication of the task. The order in which the cells are undertaken is separately randomized for each subject.

The experimental variables are (1) viewing condition, (2) transmission delay, and (3) type of computer augmentation, as indicated in Figure G-1. Viewing conditions are varied by allowing the subject to observe directly from a point about six feet from the table or by closing the separator curtain and turning on a closed-circuit TV. Audio cues are provided in both cases. Transmission delays from 0 to 10 seconds are provided using the 30-Hz delay line simulation described in Section II.

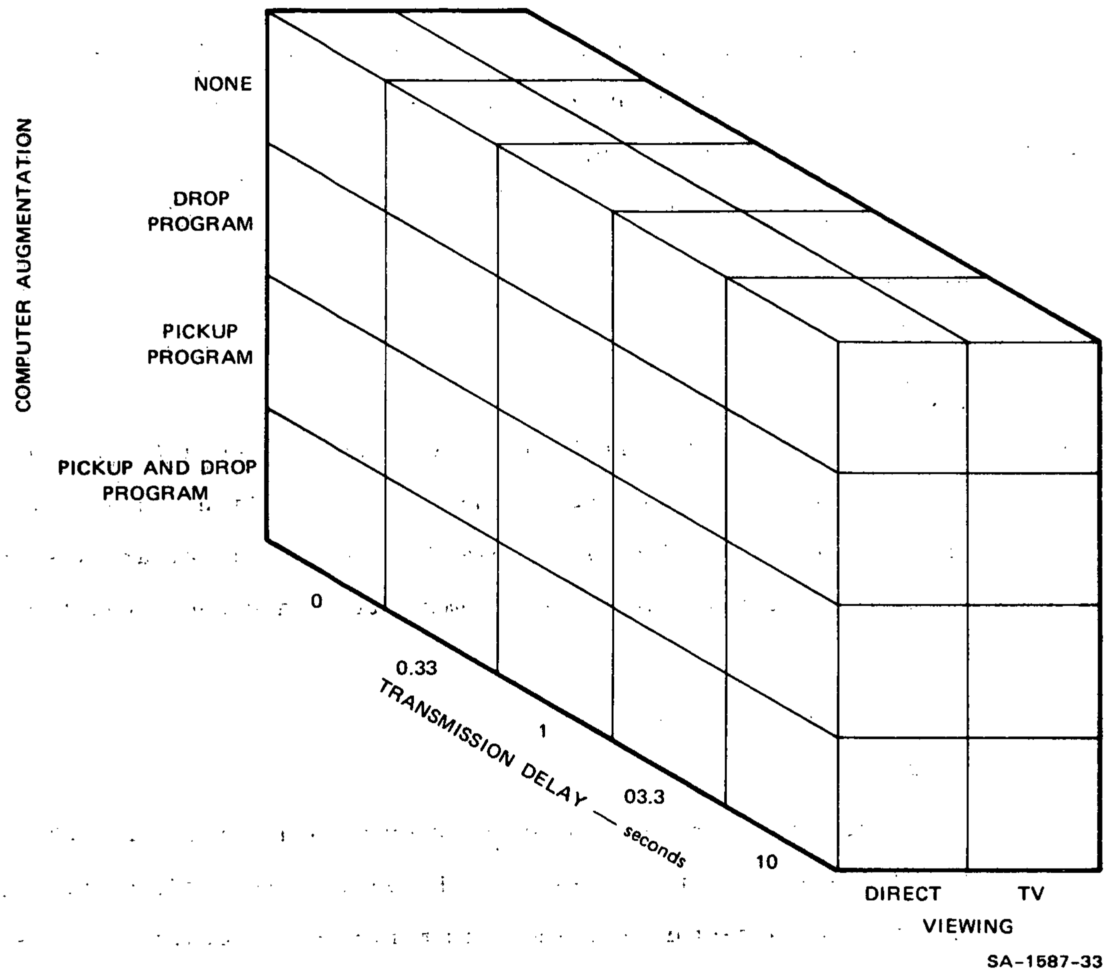


FIGURE G-1 DESIGN FOR SUPERVISORY CONTROL EXPERIMENT

Computer augmentation is provided by three ARM programs for automatically controlled manipulation based on touch sensing. The four levels of augmentation used in the experiment are:

- NONE. Only manual control is permitted.
- DROP. On command, the computer moves the grasped object to the receptacle and drops it in.
- GROPE. On initial touch contact with the object, the computer directs the arm to automatically move it in closed jaws.
- GROPE and DROP. Automatic entry to DROP after GROPE is completed.

Multiple measurements on each picking up attempt are made by the performance monitoring system described in Section IV. The measurements important to this experiment are:

- Task time
- Power consumed
- Number of moves
- Move-time histograms.

Characteristics of the move-time histograms, such as time delay and a mathematical model coefficient describing the distribution, should also be obtained. Useful move-time histograms for the experiment are master move time, slave move time, total measurement time, and human reaction time.

3. Analysis

Each experimental variable would be entered in a 4 X 5 X 2 analysis of variance to determine its statistical characteristics and, hence, its ability to serve as a performance index in manipulation evaluation experiments. The variance analysis would also show how and with what significance levels the experimental conditions affected the individual variables. Hopefully, there will be a substantial reduction in task time (one of the most important operational variables) with increasingly automatic operation.

Appendix H

TESTS AND ACTIONS

Appendix H

TESTS AND ACTIONS

This appendix lists the 19 tests and 25 actions of the ARM languaɩ Descriptions of these instructions as they affect the simulated remote computer structure previously shown in Figure 6 are broken down into eight categories.

1. General Single-Word Tests

- DONE--Passed if all the components of the error vector are less than EPSILON. EPSILON may be changed by the instruction SET EPSILON; X. The DONE test indicates whether or not the arm has finished moving to a new command position.
- WAIT--Passed if the timer is not yet zero. The timer may be set by the instruction SET CLOCK; X.
- JAW--Passed if the jaw opening is greater than JSIZE, which is changed by the instruction SET JSIZE; X. Different actions can be carried out, depending on the size of the grasped object.
- ENDLOOP--Passed if the LOOPCOUNT is zero. Each time this instruction is executed, the LOOPCOUNT is decremented by one and then tested. The LOOPCOUNT may be set by the instruction STARTLOOP; X.

2. Single-Word External Sensor Tests*

- FINGERBOTTOM--Passed if either of the two sensors on the bottom of the index finger is on.

---

*See also Appendix B.

- THUMBBOTTOM--Passed if either of the two sensors on the bottom of the thumb is on.
- ANYSENSOR--Passed if one or more of the external sensors are on.
- WEBB--Passed if the webb sensor at the base of the end effector between the thumb and index finger is on.
- TOPSENSOR--Passed if any external sensor on the top of the hand is on.
- BOTTOMSENSOR--Passed if any external sensor on the bottom of the hand is on.
- FRONTSENSORS--Passed if any front, or finger tip sensor is on.
- INDEXSIDE--Passed if either of the two sensors on the side of the index finger is on.
- THUMBSIDE--Passed if either of the two sensors on the side of the thumb is on.
- EXMASK--Passed if any one of the external sensors specified by a set of two mask words is on.

3. Single-Word Internal Sensor Tests*

- GRAB--Passed if more than THRESH sensors of the 288 sensors (internal sensors) on the finger pads are on. THRESH is changed by the instruction SET THRESH; X.
- ROWTEST--Passed if more than THRESH internal sensors are on. The rows tested are determined by a row mask word.
- COLUMNTEST--Passed if more than THRESH internal sensors are on. The columns tested are determined by a column mask word.
- ROWDIFF--Passed if more than THRESH internal sensors are on and if the number of sensors on in the rows tested by the mask word is greater than the number of sensors on in the rows not tested by the mask word.

---

*See also Appendix B.

- COLUMNDIFF--Passed if more than THRESH internal sensors are on and if the number of sensors on in the columns tested by the mask word is greater than the number of sensors on in the columns not tested by the mask word. Note that the use of masks allows specific areas of the jaw to be tested for contact and compared with others.

4. Single-Word Actions

- AUTO--Causes the analog commands switch to open so that the command register can be used for automatic operations.
- CLOSE--Causes the jaws to close by placing the closed jaw angle in the proper entry of the command register.
- OPEN--Causes the jaws to open by placing the open jaw angle in the proper entry of the command register.
- STOP--Causes the contents of the position register to be placed in the command register, stopping the arm.
- TRANSFER--Subtracts the contents of the commands register from incoming analog commands, places the result in the transfer register, and closes the auto-manual switch. This allows control to be taken over externally with no transient motion.
- MANUAL--Causes the analog commands switch to close so that the command register is used for analog signals from the operator.
- REPEAT--Increments program pointer by one.
- SKIP--Increments program pointer by one.
- SKIP 2--Increments program pointer by two.
- DDT--Causes the program to enter the DDT mode.
- BELL--Causes the teletype bell to ring.

5. Two-Word Actions

- SET CLOCK; X--Loads the timer with the contents of the next memory cell (timer set to X). The number in the timer, which is reduced by one every 30th of a second, can be tested to limit the length of a move.

- GO TO; X--Loads the program pointer with X so that instruction X will be the next one executed.
- START LOOP; X--Loads the loop counter with X.
- SET THRESH; X--Sets the internal sensor threshold.
- SET EPS--Sets the tolerance within which the arm position must be maintained so that the DONE test will be passed.
- SET RMASK; X--Sets the internal sensor row mask word, which determines what areas of the sensor pads will be tested. The rows run the length of the jaws in parallel fashion. There are six rows per pad, and twelve runs total. Thus, a single bit in the row mask word tests a single row on one of the jaws (see Appendix B for mapping).

6. Two-Word Vector Actions

- DEFINE; X--Contents of the position register are stored in memory, starting at location X. This stores the set of joint angles (a posture or, equivalently, a position of the end effector in space) to be returned to later. X may also be a maximum six-letter name, i.e., BOX, BIN, etc.
- MOVE TO; X--The contents of the position register are replaced by the contents of memory starting at location X, thus causing the arm to assume the new joint angles. X may also be a maximum six-letter name.
- INCREMENT BY; X--The contents of memory starting at location X are added to the contents of the command register.
- DECREMENT BY; X--The contents of memory starting at location X are subtracted from the contents of the command register.
- SET GAINS FROM; X--The contents of the gain register are replaced by the contents of memory starting at location X. Reducing a particular gain makes the joint more "spongy"; setting a gain to zero makes the joint free to move or comply with external forces.

7. Three-Word Actions

- SET EMASK; X_1; X_2--Sets the two external sensor mask words to X_1 and X_2. Each bit position specifies a specific external sensor to be tested. The mapping of bits to sensors is given in Appendix B.

- SET CMASK; X_1; X_2--Sets the two internal sensor column mask words to X_1 and X_2. These mask words determine what areas of the sensor pads will be tested. The columns run the width of the jaw in parallel fashion. There are 24 columns, 12 in the front half and 12 in the rear half of the pads. Thus, each bit of the two 12-bit words tests a single column on the jaws. The mapping of bits to columns is given in Appendix B.

8. Four-Word Actions

- RUN AUTO; X_1; X_2; X_3--This response causes an ARM program to be loaded into the program area and run. Thus, one ARM program may call another. The values of X_1, X_2, and X_3 are the six-bit universal random character set (SBURCS) representations of the name of the program. X_1 is the representation of the first two letters, X_2 the second two, and X_3 the third two letters. Thus, each program name may be a total of six characters long.

APPENDIX I

TOUCH SENSORS AND AUTOMATIC CONTROL

Paper Presented to

The Fourth International Symposium
on External Control of Prosthetic Devices
Dubrovnik, Yugoslavia
28 August to 2 September 1972

Introduction

Commands for movement, which come down the spinal cord, are currently thought to be length commands to individual muscles, which are in a length-control servo loop (Merton, 1972). Yet it is not known how man, who perceives, thinks, and acts in terms of near, far, left, right, away from, toward, and so on, communicates with his muscles to achieve desired results in these terms. By studying how man does things, and trying to synthesize the sequence of control and sensing actions he takes, we can simulate, experiment with, and perhaps eventually understand his communication with his muscles. Such an approach may allow us to determine what sensory information should be measured and reproduced for the human using a teleoperator or prosthetic device, to make his job more realistic and allow him to project himself into his work. Such information is the key to determining what types of movements and an effector should be capable of making, and what type of sequential organization is necessary for supervisory control and automatic programming.

An example of an analysis of actions is illustrated in Figure 1. Here it is assumed that a man has accidentally dropped a coin on the ground, and the figure shows his actions as he reaches down to pick it up. The analysis illustrated in Figure 1 is roughly the following:

(1) Bending over, he puts his hand in the pincher grip position with tumb and index finger opposing. The remaining three fingers are gently clenched or sometimes fully extended.

(2) Reaching down, he feels unstable; he stabilizes himself with his thumb (resting some weight on it), picking up the coin with the index finger.

(3) His thumb touches the ground--Figure 1(a).

(4) He continues pushing until his index finger touches the ground--Figure 1(b).

(5) He slides his index finger on the ground toward his thumb until the coin is felt against the thumb--Figure 1(c).

(6) Maintaining a finger-thumb squeeze, he lifts the index finger and rotates the coin--Figure 1(d).

(7) When the rim of the coin touches his thumb, he applies more closing force and lifts his hand.

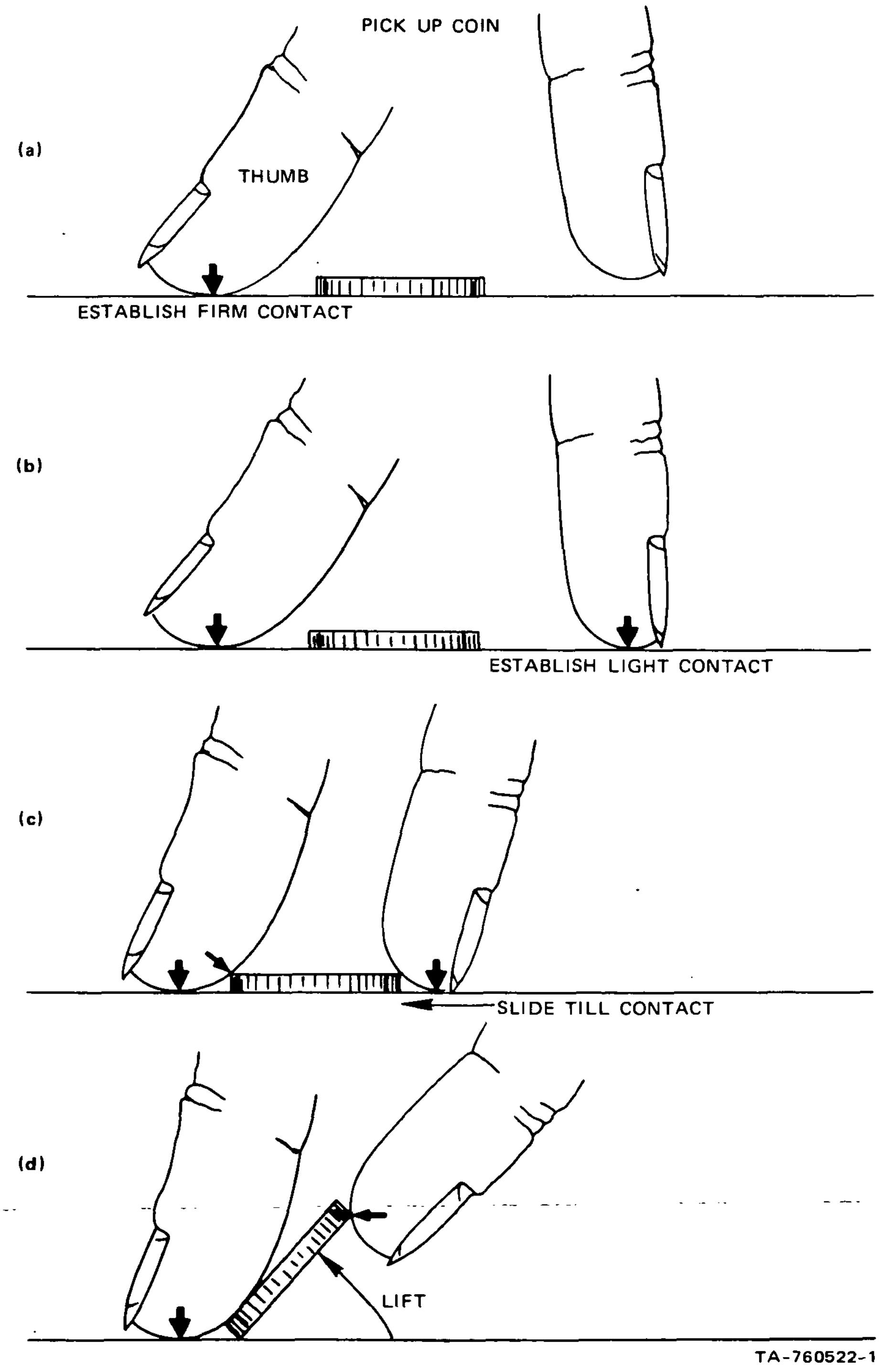


FIGURE 1 ANALYSIS OF HAND PICKING UP COIN

This analysis tells us that, to follow this plan with a teleoperator or with automatic control, we need touch information from several areas on the hand. Similarly, an end effector is needed with ability to retract the index finger, as well as the normal ability to close index finger against thumb.

A sequential breakdown sets a series of constraints on an automatic control system designed to carry out the task. The controller must be able to:

(1) Maintain fingertip contact while closing prosthetic fingers.

(2) Close the fingers until the thumb is touched.

(3) Raise and close the index finger around a natural hinge point.

(4) Detect when an object is safely gripped.

A general purpose arm controller must not only be able to establish a wide variety of control loops on the basis of touch and position information but also be able to carry out a sequence of these control loops by branching from one to the next when certain completion criteria have been met. For the coin pickup example, the flow chart of Figure 2 was prepared to show the sequence of steps an arm automaton must follow through.

After a range of such tasks has been analyzed, it becomes apparent that tactile information and sequenced automatic control play important roles in manipulation. Therefore, the three steps below are suggested as a general strategy for picking up an object.

(1) A man looks at an object, and a sequence of actions and reflex-like responses takes form in his mind. (Included in his plan are a desired initial and final hand--or arm--posture. The plan may even be formed during the first movement toward the object.)

(2) He puts his hand in the desired initial posture.

(3) He initiates a motion toward the object and reflexively carries out his plan of acquisition, using touch, force, and position information as required.

In this strategy, tactile information is used for alignment, e.g., for positioning an object in the hand or locating exactly where an object

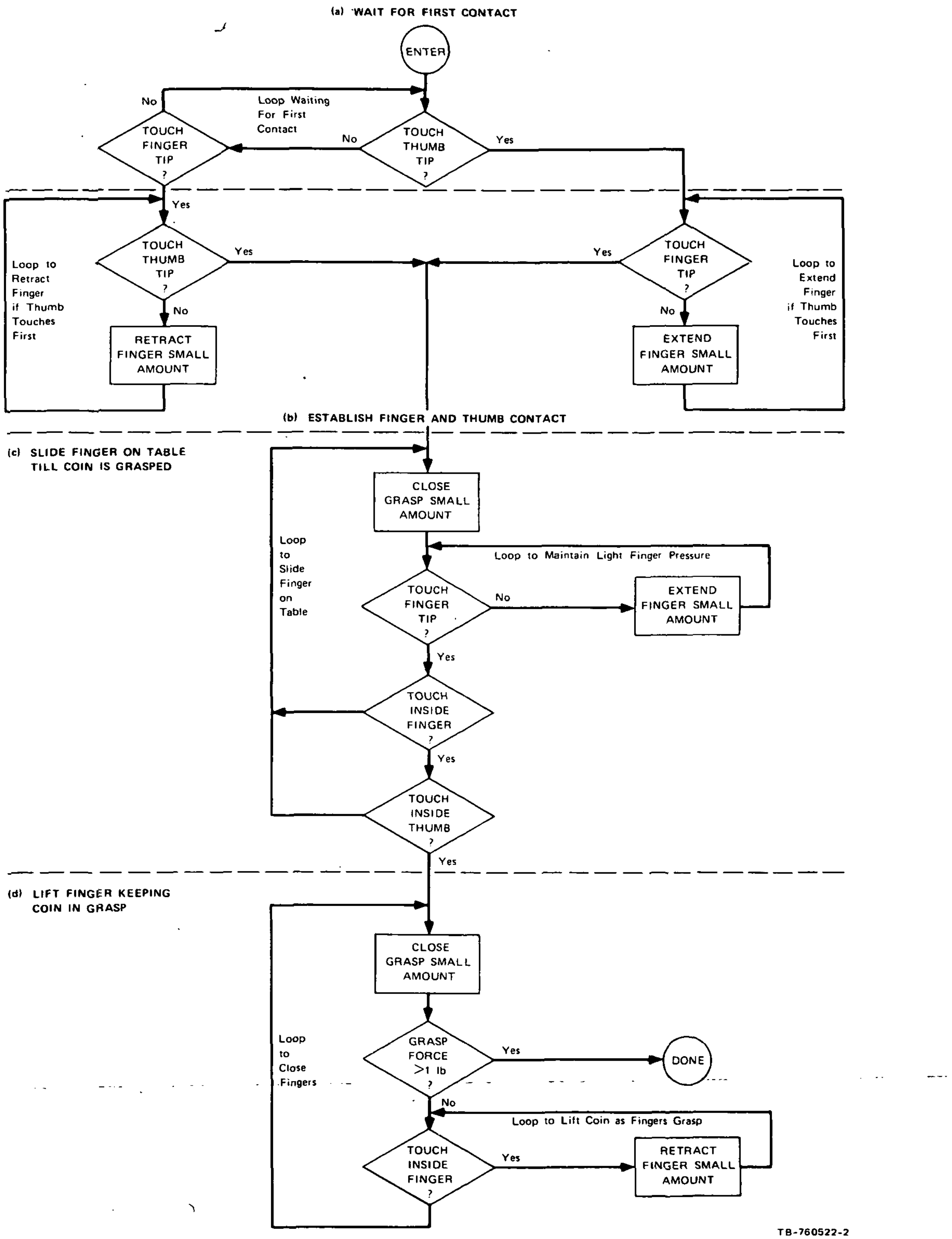


FIGURE 2 FLOW CHART FOR HYPOTHETICAL PROGRAM TO PICK UP COIN FROM TABLETOP

is and for limitation of the forces applied to an object. Sequenced automatic control is evident in our every practiced move; we can unscrew a nut from a bolt, knit, or write our signature, all quickly and without looking. The combination of programmed motions and touch sensing permits a wide range of tasks to be carried out automatically (Tomovic, 1964).

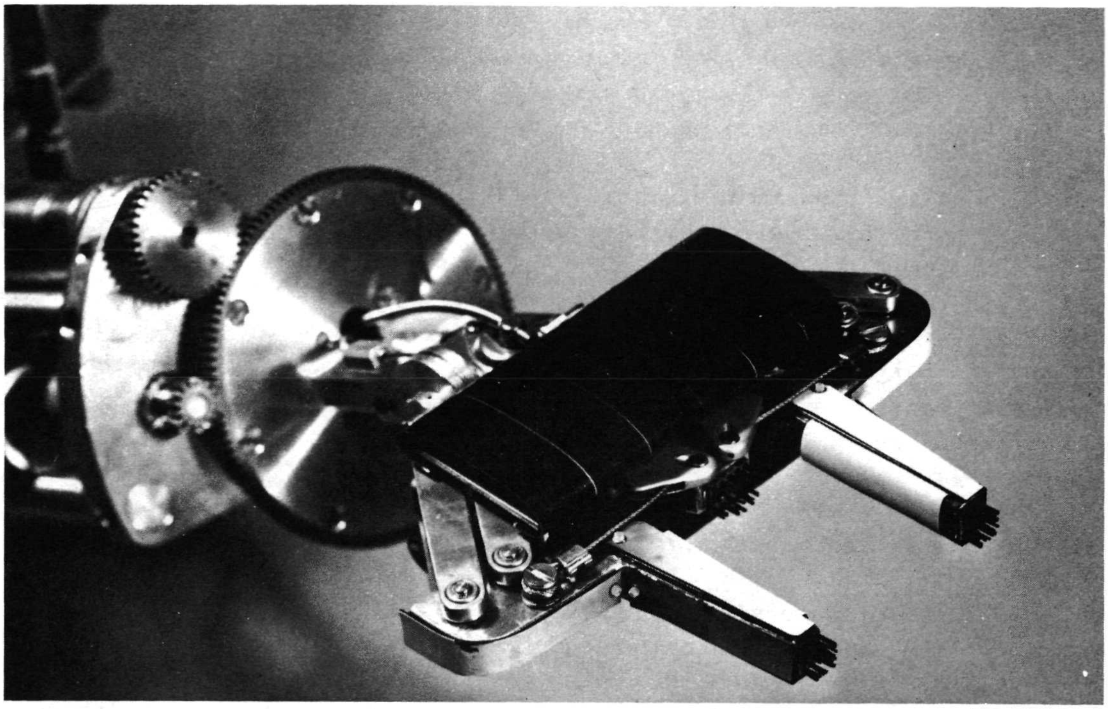
These two important elements of man's skill and ability to manipulate are not normally supplied by teleoperator control systems. Following the approach suggested at the beginning of this paper, we have experimented (1) with tactile feedback by incorporating a touch sensing and touch display system into a teleoperator and (2) with different levels of automatic control by incorporating supervisory control features in the teleoperator control system. The concepts on which these experiments have been based are explained in the following four sections of the paper.

Touch Sensing and Display

One way of understanding the contribution of touch sensing to man's manipulative skills is to provide a teleoperator with different kinds of feedback touch from slave to master for teleoperator control. Carrying out manipulations with the touch feedback on or off, we can determine what skills the new information has provided. After identifying a skill in this way, we also have an existence proof that the same skill can be obtained from an automatic control system utilizing the same tactile information.

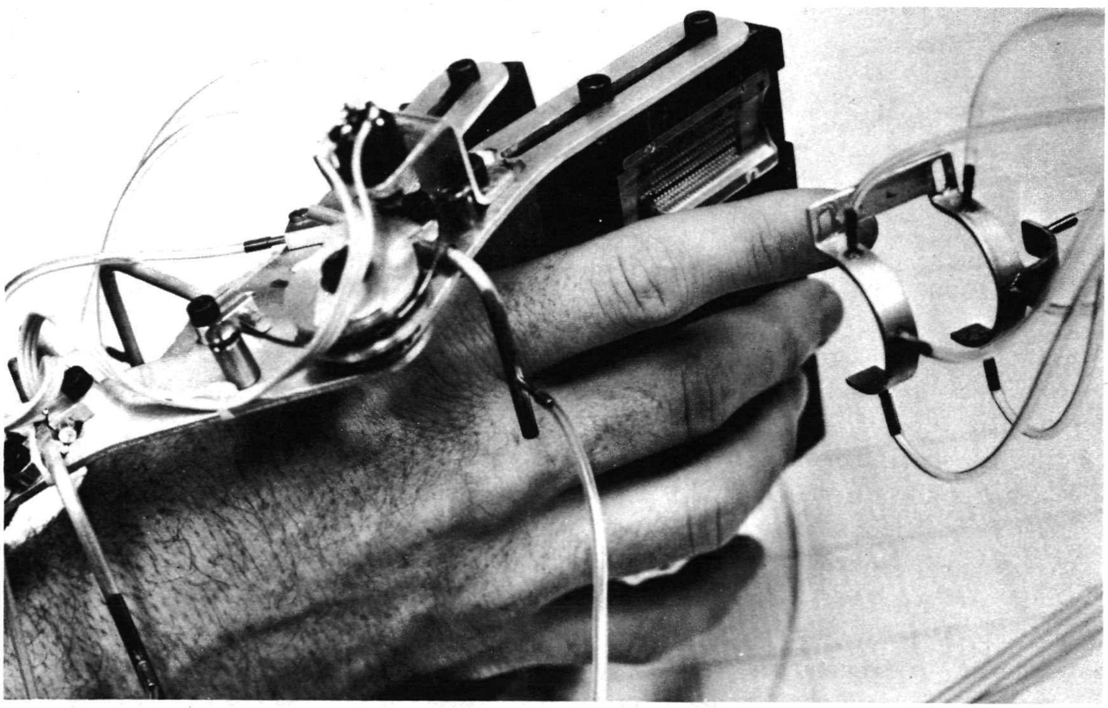
Two touch-feedback systems for the teleoperator control system have been constructed for experimental evaluation. Each system consists of a set of sensors mounted on a mechanical arm and a corresponding set of tactile stimulators mounted on a control brace. All the sensors are constructed of conducting rubber that is deformed to complete an electrical circuit upon physical contact (see Hill and Bliss, 1971, for construction details). The construction of most of them depends on etched wiring on printed circuit boards. Individual sensors activate corresponding stimulators in a binary fashion: A stimulator is either full on or full off.

The hand contact system senses and reproduces to the operator the contact between the end effector and the object being touched or manipulated. This system consists of a number of conducting rubber sensors mounted on the outside surfaces of the mechanical hand, as shown in Figure 3. The tongs of the hand are completely covered with these sensors (seven sensors per tong), as are the extreme or protruding parts of the upper hand (seven sensors). The sensors are so arranged that any contact of the hand with a flat surface is sensed and that any contact with the tongs is sensed. Each sensor is connected via amplifying and gating circuits to an air-jet tactile stimulator. The air jets are positioned on the control brace to



TA-760522-4

FIGURE 3 CONFIGURATION OF TOUCH SENSORS



TA-760522-5

FIGURE 4 TACTILE STIMULATORS ON THE HAND CONTROLLER

produce touch sensations on portions of the operator's hand corresponding to the locations of the sensors. Each jet produces an area of pulsating pressure on the skin approximately 3/16 inch in diameter. The arrangement of air-jet stimulators on the control brace is shown in Figure 4. The construction of the air-jet stimulators was described by Bliss and Crane (1965). By using the external sensory feedback system, it is possible to (1) reach into a box without the aid of vision and extract a block from it, (2) locate a visually obscured object to be picked up under the tongs and respond by grasping it, and (3) control wrist rotation so that both tongs rest on a flat surface.

The jaw contact system senses and reproduces to the operator the shape and location of the object held in the remote jaws. Two sensing pads are built into the tongs of the mechanical hand (as shown in Figure 3). Each of these two opposing pads consists of 144 individual contacts in a 6 by 24 rectangular pattern. Two corresponding 6 by 24 rectangular arrays of bimorphs contacting the index finger and thumb are built into the control brace, as shown in Figure 4. Bimorphs produce a 250-Hz vibration of the skin, restricted to an area about 1 mm in diameter. Thus the pattern of contact closures is reproduced as a pattern of vibration, enabling the operator to feel on his own thumb and index finger the shape and location of the object held in the remote jaws. A complete description of similar bimorph arrays used in shape recognition and reading experiments has been given by Bliss (1969) and Bliss et al. (1970). By using the jaw shape-sensing system it is possible to (1) pick up an object in the desired part of the tongs, as in the center or at the tip, (2) obtain rotational alignment to a fixed object while gripping it, (3) detect slippage when lifting an object and close the jaws until the slippage has stopped, and (4) pick up an egg without breaking it.

An Automatic Controller for Arm Automata

From analyses of such manipulation tasks as that of picking up a coin, it is evident that a great deal of flexibility is required in experimenting with arm control modes. Because of this requirement the entire automatic controller has been simulated, using our laboratory computer (a LINC-8 with 8192 words of memory). The simulation has separate programs for inputting manual control information, performing servo-control functions, and carrying out automatic arm control. These three programs are sequentially serviced 60 times a second. During each 1/60-second epoch, manual control inputs (if any) are accepted, and a new instruction is formed for the automatic controller. Next, the servo calculations are carried out, and, finally, the automatic processor is allowed one sequential operation based on an instruction. The chief advantages of this simulation are the flexibility and speed with which the control structure

and parameters can be included or modified through program (software) rather than equipment changes. The 60-Hz sampling rate has been found adequate for preserving human accuracy and smoothness of operation.

A block diagram of the automatic controller is given in Figure 5. The instruction (two 12-bit words) and the analog joint commands (seven 12-bit words) from the control station are the only inputs. The auto-manual switch is under program control and can be either closed to accept manual inputs from the operator or open to allow commands generated by programs to move the arm. Arm control is quite conventional, with actual joint positions (obtained by analog-to-digital conversion) subtracted from the command joint positions, and the difference multiplied by the joint gains and then output to the servo motors (via digital-to-analog conversion) to establish angular rates. The transfer register is used to offset the analog commands, so that control can be "transferred" to the human operator smoothly after an automatic operation has been completed.

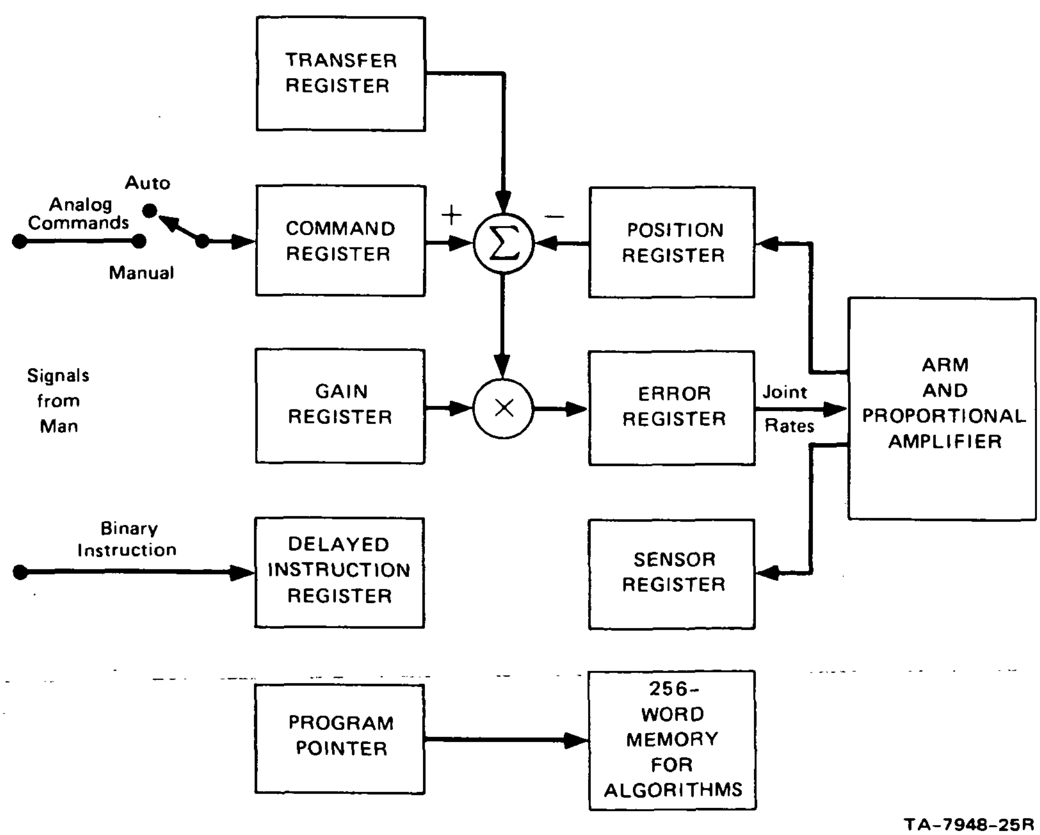


FIGURE 5 COMPONENTS OF THE COMOUTER CONTROLLER

Discrete instructions transmitted by the human operator are saved in the instruction register. The basic form of the instruction specifies a test, an action, and a numerical parameter. The 256-word memory contains sequential lists of instructions for carrying out manipulation tasks, and the program pointer indicates where the next instruction from the memory list will be found.

Not shown in Figure 5, but essential to the operation of the automatic controller, is the instruction processor shown in Figure 6. The processor transfers numbers between registers and carries out sensor and position tests on the basis of individual instructions. These instructions are the building blocks used by both single commands from the control typewriter and multiinstruction programs of manipulation.

A single instruction requests that a specific test be executed and that a specific command be carried out if the test is passed. The first half of the instruction word (6 bits) is used to select one of 64 possible tests by means of a look-up table. If the test is passed, the second half of the instruction (6 bits) is similarly used to select one of 64 possible actions. Even though only 19 tests and 25 actions have been implemented, a rich variety of operations is already possible. An example of a single instruction is:

If <u>fingertip</u> <u>sensor</u> <u>closed</u>, then <u>open</u> <u>jaws</u>.

For the computer, this instruction is coded by FO, where the <u>fingertip</u> <u>sensor</u> <u>closed</u> test is specified by F and the <u>open</u> <u>jaws</u> command is specified by O. Table 1 lists representative tests and actions.

Algorithmic Language for Remote Manipulation (ARM)

If requests for the automatic operations described in the preceding section are taken from a list, the list can be considered a program of motions (an algorithm) to carry out a manipulation task. The effective utilization of such a program, however, requires a means for writing it in an easy-to-use language and a means for assembling (or generating) a list of arm operations from the statements in the language. Under the constraint of a small computer system, we simultaneously developed the separate concepts of the ARM language, the assembler, and the instruction set for the automaton controller previously described.

ARM is an extension of the MHI or THI language developed by Ernst (1961) and of MANTRAN developed by Barber (1967), in that manual inputs from the operator can be used in addition to teletype inputs. Thus, the

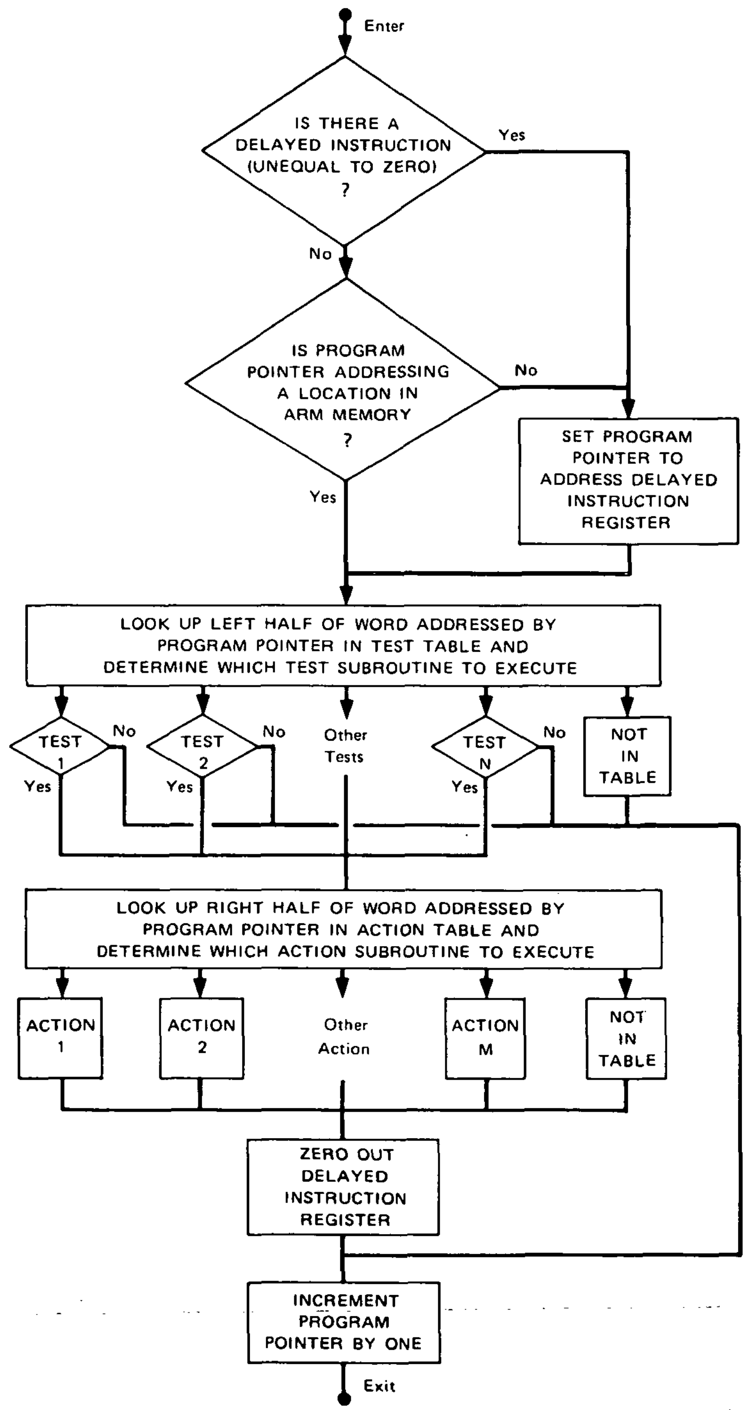


FIGURE 6 INSTRUCTION PROCESSOR. This subroutine carries out single instructions and programs of instructions. It can be seen that, if instructions are being taken from a program in arm memory, an instruction sent from the control station will cause the program to be stopped and the instruction to be carried out.

Table 1

REPRESENTATIVE TESTS AND ACTIONS

Tests

Finger Bottom--Either of the two sensors on the bottom of the index finger is on.

Anysensor--One or more of the external contact sensors on the end effector are on.

Grab--More than "Sthresh" sensors of the 288 sensors on the finger pads are on. Sthresh is changed by the instruction "Set Sthresh; X."

Done--All the components of the error vector are less than "Epsilon." Epsilon may be changed by the instruction "Set Epsilon; X." The done test indicates whether the arm has finished moving to a new command position.

Jaw--The jaw opening is greater than "Jsize," which is changed by the instruction "Set Jsize; X." Different actions can be carried out, depending on the size of the grasped object.

Simple Actions

Auto--Causes the analog commands switch to open so that the command register can be used for automatic operations.

Close--Causes the jaws to close by placing the closed jaw angle in the proper entry of the command register.

Stop--Causes the contents of the position register to be placed in the command register, stopping the arm.

Transfer--Subtracts the contents of the commands register from incoming analog commands, places the result in the transfer register, and closes the auto-manual switch. This allows control to be taken over externally with no transient motion.

Table 1 (Concluded)

Set Timer; X--Loads the timer with the contents of the next memory cell (timer set to X). The number in the timer, which is reduced by one every 60 second, can be tested to limit the length of a move.

Go To; X--Loads the program pointer with X so that instruction X will be the next one executed.

Vector Actions

Define; X--Contents of the position register are stored in memory, starting at location X. This stores the set of joint angles (a posture, or, equivalently, a position of the end effector in space) to be returned to later.

Move to; X--The contents of the position register are replaced by the contents of memory starting at location X, thus causing the arm to assume the new joint angles.

Increment by; X--The contents of memory starting at location X are added to the contents of the command register.

Set gains from; X--The contents of the gain register are replaced by the contents of memory starting at location X. Reducing a particular gain makes the joint more "spongy;" setting a gain to zero makes the joint free to move or comply with external forces.

operator can move his control brace and request that the arm move to "this" position or move "this" joint "this" much. Each "this" in the preceding sentence is a manually specified quantity that is difficult to verbalize, much less to quantify as a joint vector for typewriter input.

An example of a program (written in ARM), the algorithm for the actions of picking up a coin previously described, is given in Table 2. An entire program is given to an assembler for conversion to a list of numbers (instructions) for execution by the remote computer. Compiling is quite straightforward: Values for the various symbols on each line are simply added together to form the instruction.

Even after the pickup program of Table 2 has been loaded into the controller's memory and started, the grasping sequence is not begun until the index finger or thumb of the end effector has been brought into contact with an object. After the sequence of moves has been finished, the program returns smoothly to the manual control mode by the transfer command. If for any reason the operator wishes to shop in mid-task, he need only transmit any one instruction to the controller.

As can be seen from the coin pickup program (Table 2), the language is quite simple and powerful, needing only 42 12-bit instructions and two command vectors for carrying out the coin pickup. The longest ARM program written to date uses 60 instructions and 42 storage locations to direct a 7-degrees-of-freedom manipulator to unscrew a nut from a bolt and deposit it in a receptacle (Hill and Bliss, 1971). The length of this program could have been halved if inclusion of special procedures had not been necessary to compensate for inprecision in the manipulator and control system used.

Supervisory Control

Our goal is to design a control scheme that optimizes performance in carrying out remote tasks by combining the best attributes of man and computer. Therefore, man's ability to interpret scenes, estimate distances, and project motion with a multicoordinate control brace is combined with the computer's ability to save and accurately duplicate arm positions, remember sequences of motions, carry out tests based on arm position, and interpret touch sensors. General background material on such supervisory control has been given by Johnson and Corliss (1967) and Corliss and Johnson (1969).

The supervisory control scheme for interacting with the previously described automatic controller is governed by the command tree shown in

Table 2

COIN PICKUP PROGRAM

```
BEGIN

WAIT:       IF THUMBTIP GO TO; EXTEND            /Loop Waiting for First Contact
            IF FINGERTIP GO TO; RETRACT
            GO TO; WAIT

EXTEND:     IF FINGERTIP GO TO:  SLIDE           /Loop to Extend Finger if Thumb
            INCREMENT BY; DX                     /Touches First
            GO TO; EXTEND

RETRACT:    IF THUMBTIP GO TO; SLIDE             /Loop to Retract Finger if Finger
            DECREMENT BY; DX                     /Touches First
            GO TO; RETRACT

SLIDE:      DECREMENT BY:  DOPEN                 /Slide Finger on Table Till

S1:            IF FINGERTIP GO TO; S2            /Coin is Grasped
               INCREMENT BY; DX
               GO TO S1

S2:         IF INSIDE FINGER SKIP 2
            GO TO; SLIDE
            IF INSIDE THUMB GO TO; LIFT
            GO TO; SLIDE

LIFT:       DECREMENT BY; DOPEN                  /Lift Finger Keeping Coin in Grasp
            IF JAWCLOSED TRANSFER                /Return Control to Man

L1:            IF INSIDE FINGER SKIP 2
               GO TO:  LIFT
               DECREMENT BY; DX
               GO TO; L1

DX:         1; -1; 0; 0/Incremental Command Vector to Extend Index

DOPEN:      0, 0, 0; -1/Incremental Command Vector to Open Fingers

END
```

Figure 7. Moving from branch to branch by typing single letters, numbers, or short names, the human operator can specify communication options from three increasingly automatic levels of control. After the operator has typed a character on a keyboard, specifying which control branch he wishes to take, the computer first prints out one of the short messages indicated in Figure 7 by quotation marks; it then prints, on a new line, the prompting symbol for the new control branch. This approach limits the amount of coded information required to carry out a task, permitting use of a typewriter keyboard with one hand and a control brace or joystick for manual control with the other. A simple five-finger binary keyboard or even a telegraph key might be more satisfactory than a typewriter keyboard!

The first, or manual, mode of control is obtained by typing K (for knobs, a bank of potentiometers), B (for a 7-degrees-of-freedom Rancho control brace worn on the right arm), or T (for teletype), followed by A (for absolute) or R (for relative) Absolute control causes joint positions to be read directly from these devices and to be transmitted to a remote station. Relative control specifies that joint commands from the control source take up where the previous joint commands left off. Thus, after a relative transfer, the new control source continues where the old one left off, and there is no transient motion artifact.

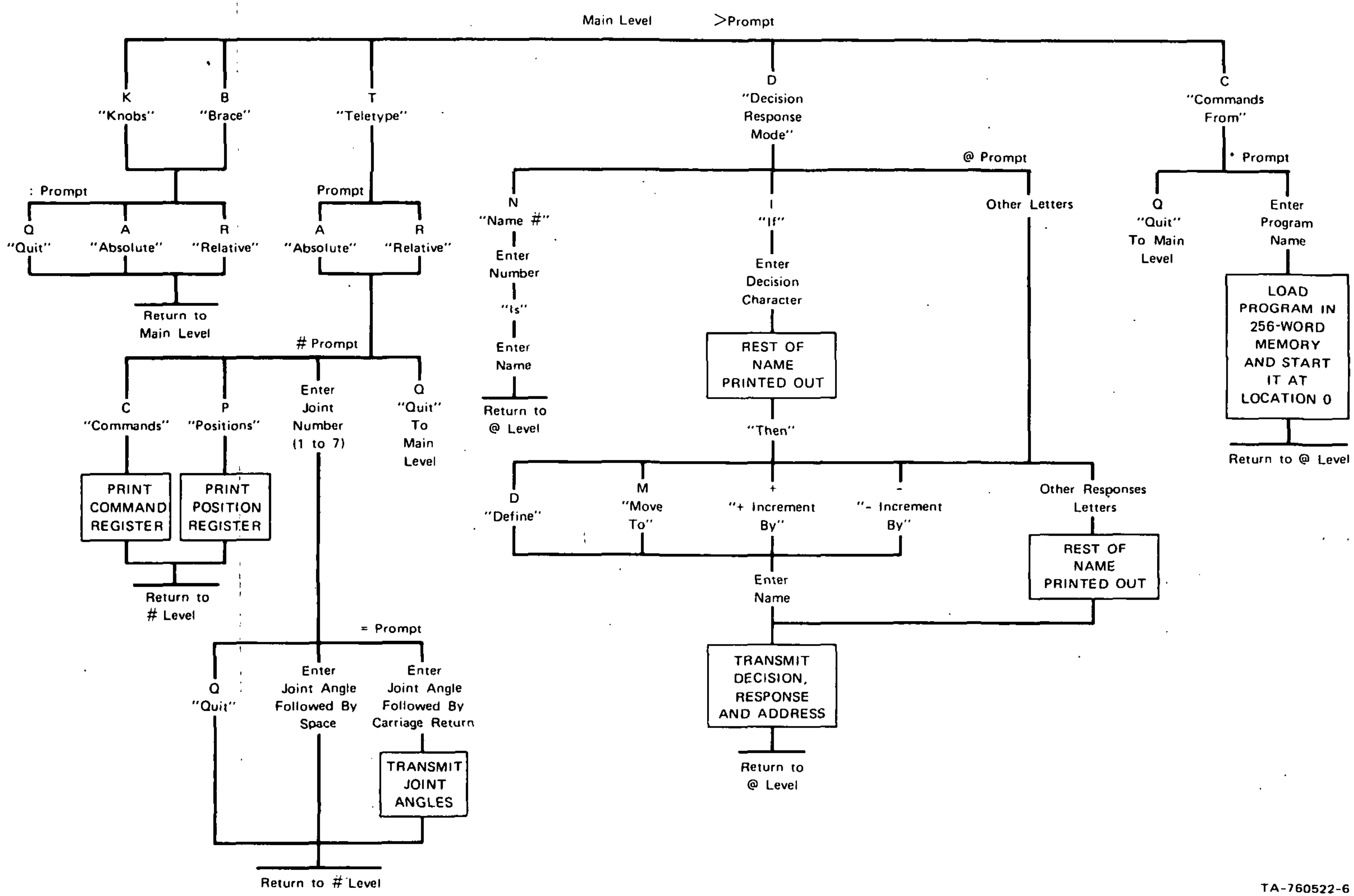
The second, or decision-response, mode of control is obtained by typing D; it permits instructions to be transmitted to the arm automation. If I is typed, the computer prints IF and is prepared for a character identifying a test; otherwise the program is prepared for a character identifying an action. For example, Test T is "thumb sensor closed," and Action O is "open jaws." Therefore, as a sequence I, T, O is typed, the instruction "If Thumb then Open" is printed out, coded, and transmitted to the automaton (manually entered letters are underlined here). In conjunction with the vector actions (Define, Move to, +Increment by, and -Increment by), it is possible to use names previously entered that correspond to addresses of joint-angle vectors in memory. Thus, after entering the name BOX in the catalog, it is possible to enter the commands:

Define BOX

Move to BOX

If thumb then move to BOX.

All of the 19 tests and 26 actions built into the automatic controller are thus executable by typing a few characters under decision-response control.



TA-760522-6

FIGURE 7 SIMPLIFIED COMMAND TREE FOR THE SUPERVISORY CONTROL STATION

The third, and most automatic, mode of supervisory control is obtained by typing C (Commands from*) followed by the name of a program previously constructed by using ARM language. This causes the program of tests and actions to be loaded into the remote computer's 256-word memory. The supervisory controller then switches the operator to the decision response mode for further interaction with the program. Positions of objects the program needs to know about are input by using the Define command. This command causes the seven joint angles of the arm to be stored at the program location associated with the name in the catalog. These named vectors are stored in a region outside the 256-word memory and are accessible to any program run.

Currently, a series of ARM programs are being developed to perform a variety of useful tasks based on remote information from touch sensors and joint angles. These programs control the arm to unscrew a nut from a bolt, search for an object on a tabletop, align the jaws with a fixed object, center an object in the jaws horizontally and vertically, hold an object lightly without letting it slip, and yield to an external force. When a working library of these programs becomes available, a fourth, and more useful, mode of automatic control can be constructed, to link ARM programs as subroutines.

Applications to Prosthetics Control

Although the work described in this paper has been concerned with control of a mechanical arm, the advantages of sensory feedback and automatic control are also applicable to the design of prosthetic devices. As an example, consider the problem of controlling a prosthetic hand with the 24 degrees of freedom of the human hand. This end effector would have 24 motors pulling on 24 cables connected to the points on the model hand corresponding to the points where the tendons of the human hand anchor. Obviously, it is impossible to control the 24 degrees of freedom directly by using separate myoelectric potentials from the amputee's stump, where there may only be one such analog signal available.

Instead of using one-to-one analog control on 24 channels, a better way to solve this control problem would be to use one analog channel and one channel for coded instructions. The analog channel would be conventional, with myoelectric potentials used to provide a variable control signal. The instruction channel would convey information to the prosthesis controller governing the use of the analog channel. The source of the discrete, coded information could be a single free muscle in the body that could communicate information such as characters of the international telegraph code. Through a series of dots and dashes, individual letters

and numbers could be transmitted to a prosthesis control computer. Such a hand-prosthesis supervisory control system is shown in Figure 8.

A series of examples will most easily explain this approach. The code "dash-dot" could signify that the prosthesis controller should give commands to extend the index finger and thumb and commands to curl up the remaining fingers. Additionally, the computer would interpret the single analog signal as the control source to move the thumb and index finger in opposition to form a pincher grip. We would identify this dash-dot symbol as the pincher grip instruction. Other similar control modes for grasping a handle, pencil, ball, and so forth could be set up upon receipt of particular instructions. One format for governing these gripping procedures is:

- Establish an initial posture.
- Choose an incremental control vector that, when added to the 24-element command vector, causes the end effector to produce the correct movement.
- Set the gains register to fix or free (whichever is appropriate) the degrees of freedom that do not change.
- Use the analog signal to control repetitive addition or subtraction of the new incremental vector and the control vector.

A parallel can be drawn between manual control with the supervisory arm control system previously described (knobs, brace, and so on) and manual control with a prosthetic hand. By extending this relationship, it would be possible to adapt the previous supervisory control approach to implement a "hand" decision-response mode and a "hand" programming mode. Both would be based on signals from the 24 cable-pulling servos in the hand and a network of touch sensors distributed on its surface. Because of the close correspondence of the arm and hand control systems, the implications for further development of automatic hand control will be left to the reader.

Practically, though, what about the possibility of constructing a small, pocket-sized computer controller for this prosthesis? Estimates of the size required, based on the previously described control of a mechanical arm, suggest that approximately 1,000 gates would be needed to implement the controller and that 1,000 10-bit words of memory would be needed to implement a reasonable range of automatic operations. This is roughly the complexity (35,000 transistors) of a pocket-sized electronic slide-rule calculator currently being marketed for $400.

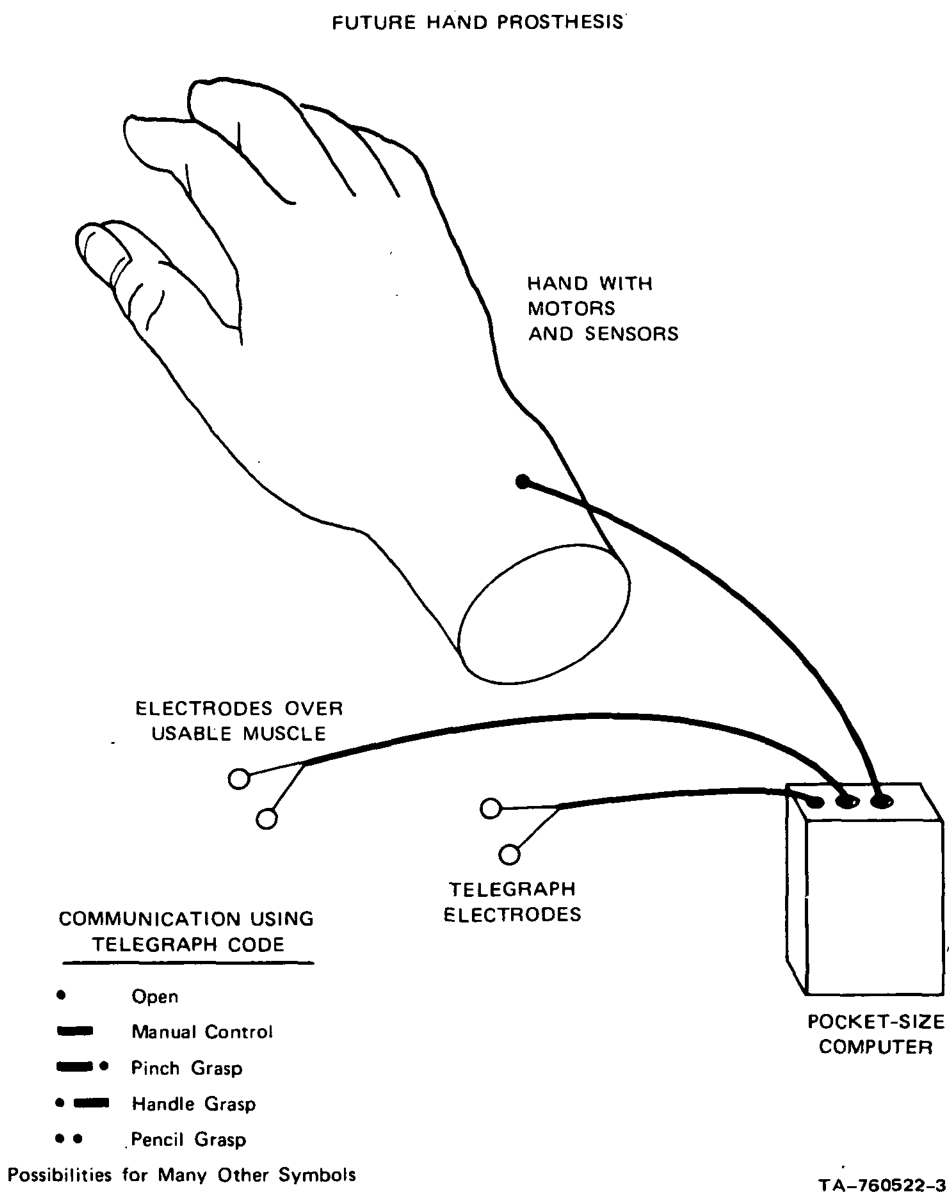


FIGURE 8 SUGGESTED CODE-CONTROLLED HAND PROSTHESIS

The problem in building this realistic hand rests, therefore, not with the automatic computer controller or with the limited manual inputs available for control, but with the mechanical design problem of building 24 servomotors of the desired power into the space available. With this supervisory control approach, the bottleneck in designing a lifelike hand prosthesis is returned to the mechanical designer.

If such a mechanical design and supervisory controller were put together, we could conceive of a man sending telegraph codes to the device and it responding by moving a pencil to print the letter corresponding to the code character sent. A single code symbol could generate a signature! This brings us back to where we began, with the problem of how man controls muscles: What kinds and what levels of commands does he use to produce the motions we see?

REFERENCES

Barber, D. J., "MANTRAN: A Symbolic Language for Supervisory Control of an Intelligent Remote Manipulator," DSR 70283-2 Engineering Projects Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts (1967).

Bliss, J. C., "A Relatively High-Resolution Reading Aid for the Blind," IEEE Trans., Vol. MMS-10, pp. 1-9 (1969).

Bliss, J. C., and H. D. Crane, "Experiments in Tactual Perception," Final Report, Contract NAS2-1679, Stanford Research Institute, Menlo Park, California (1965), Appendix B.

Bliss, J. C., et al., "Optical-to-Tactile Image Conversion for the Blind," IEEE Trans., Vol. MMS-11, pp. 58-64 (1970).

Bliss, J. C., J. W. Hill, and B. M. Wilber, "Tactile Perception Studies Related to Teleoperator Systems," Final Report, Contract NAS2-5409, SRI Project 7948, Stanford Research Institute, Menlo Park, California (July 1970).

Corliss W. R and E. G. Johnson, "Teleoperator Controls " NASA-SP-5070, Clearinghouse for Federal and Technical Information, Springfield, Virginia, pp. 7-124 (1969)

Ernst, H. A., "A Computer Operated Mechanical Hand," Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts (1961).

Hill, J. W., "Touch Sensors and Automatic Control," Fourth International Symposium on External Control of Prosthetic Devices, Dubrovnik, Yugoslavia (28 August to 2 September 1972) (proceedings in press).

Hill, J. W., and A. J. Sword, "Touch Sensors and Control," First National Conference on Remotely Manned Systems (RMS), California Institute of Technology, Pasadena, California (13-15 September 1972) (proceedings in press).

Hill, J. W., and J. C. Bliss, "A Computer-Assisted Teleoperator Control Station with Tactile Feedback," Seventh Annual Conference on Manual Control, University of Southern California, Los Angeles, California (2-4 June 1971b) (proceedings in press).

__________, "Tactile Perception Studies Related to Teleoperator Systems," Final Report 2, Contract NAS2-5409, Stanford Research Institute, Menlo Park, California (1971). CR-114346

Jex, H. R., J. D. McDonnell, and A. V. Phatak, "A 'Critical' Tracking Task for Manual Control Research," <u>IEEE Trans.</u>, Vol. HFE-7, pp. 138-144 (1966).

Johnson, E. G. and W. R. Corliss, "Teleoperators and Human Augmentation," NASA-SP-5047, Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., pp. 69-74 (1967).

Katzan, H., Jr., <u>Advanced Programming</u> (Van Nostrand Reinhold Company, New York, New York, 1971).

McRuer, D. T., D. Graham, E. Krendel, and W. C. Reisener, "Human Pilot Dynamics in Compensatory Systems; Theory, Models, and Experiments' with Controlled Element and Forcing Function Variations," AFFDL TR 65-15, DDC Document No. AD 470 337 (July 1965).

Merton, P. A., "How We Control the Contraction of Our Muscles," <u>Scientific American</u>, Vol. 226, pp. 30-37 (May 1972).

Sheridan, T. B., and W. R. Ferrell, "Remote Manipulative Control with Transmission Delay," <u>IEEE Trans.</u>, Vol. HFE-4, pp. 25-29 (1963).

Tomovic, R., "Outline of a Control Theory of Prosthetics," <u>Proc. Second Congress of International Federation of Automatic Control</u>, Basle, V. Broida, ed. (Butterworths, London, 1964).

☆U.S. GOVERNMENT PRINTING OFFICE: 1973-739-026/10

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546